NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

INTERDIGITATED BACK-SURFACE-CONTACT SOLAR CELL MODELING USING SILVACO ATLAS

by

Shawn E. Green

June 2015

Thesis Advisor:	Sherif Michael
Second Reader:	Matthew Porter

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704–0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2015	**3. REPORT TYPE AND DATES COVERED** Master’s Thesis

4. TITLE AND SUBTITLE INTERDIGITATED BACK-SURFACE-CONTACT SOLAR CELL MODELING USING SILVACO ATLAS	**5. FUNDING NUMBERS**
6. AUTHOR(S) Shawn E. Green	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

The Silvaco Atlas semiconductor modeling software was used to simulate an interdigitated back-surface-contact solar cell. The cell is modeled after the silicon-based Sunpower Corporation A-300 solar cell, which contains a number of unique features that give it advantages over conventional solar cells. This simulation attempted to match as closely as possible the results measured by the National Renewable Energy Laboratory from the A-300 cell in order to validate the model. This model was then used to investigate the effects of making the A-300 thinner, which would permit its use in military solar blanket applications. A thin and flexible solar cell is ideal for this application due to its lighter weight, making it portable and flexible, which increases its ruggedness. The ability to simulate an interdigitated back-surface-contact cell also allows future work using computer algorithms to improve power output results as well as investigations into using materials other than silicon, which may further improve power output.

14. SUBJECT TERMS Solar cell, photovoltaic device, back surface contact, Silvaco Atlas, Interdigitated, silicon	**15. NUMBER OF PAGES** 89
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

INTERDIGITATED BACK-SURFACE-CONTACT SOLAR CELL MODELING USING SILVACO ATLAS

Shawn E. Green
Lieutenant, United States Navy
B.S., University of Memphis, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2015

Author: Shawn E. Green

Approved by: Sherif Michael, Ph.D.
Thesis Advisor

Matthew Porter
Second Reader

Clark Robertson, Ph.D.
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Silvaco Atlas semiconductor modeling software was used to simulate an interdigitated back-surface-contact solar cell. The cell is modeled after the silicon-based Sunpower Corporation A-300 solar cell, which contains a number of unique features that give it advantages over conventional solar cells. This simulation attempted to match as closely as possible the results measured by the National Renewable Energy Laboratory from the A-300 cell in order to validate the model. This model was then used to investigate the effects of making the A-300 thinner, which would permit its use in military solar blanket applications. A thin and flexible solar cell is ideal for this application due to its lighter weight, making it portable and flexible, which increases its ruggedness. The ability to simulate an interdigitated back-surface-contact cell also allows future work using computer algorithms to improve power output results as well as investigations into using materials other than silicon, which may further improve power output.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
A	Amps
AR	Anti-Reflective
BSC	Back Surface Contact
CIGS	Copper Indium Gallium Selenide
DOD	Department of Defense
FF	Fill Factor
FSF	Front Surface Field
GUI	Graphical User Interface
I	Current
I_{sc}	Short Circuit Current
MPP	Maximum Power Point
η	Efficiency
NREL	National Renewable Energy Laboratory
P_{in}	Input Power
P_{max}	Maximum Power
Si	Silicon
SPACES	Solar Portable Alternative Communications Energy System
V	Volts
V_{oc}	Open Circuit Voltage
W	Watts

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Forward operating forces have equipment that requires power to operate. Supplying the resources to generate the electricity needed introduces costs and logistics burdens. Additionally, when these troops are operating away from their bases, they do not have the means to power electronic gear and must bring batteries to supply the needed energy. Batteries add considerable weight to the gear that must be carried and significantly limits the amount of time the equipment is available for use. To remedy this, there are systems available utilizing solar blankets that can be used to charge the batteries in the field, provided there is sufficient sunlight. One of these used by the Marine Corps is the Solar Portable Alternative Communications Energy System (SPACES). SPACES consist of a 0.73 m^2 solar panel and power module [1]. It is capable supplying up to 62 W of power for battery charging or to run equipment. SPACES is relatively lightweight and is rugged enough to survive the abuse of an operational environment. Part of the ruggedness of the system comes from the flexibility of the solar panels, which make them less prone to damage. It also makes them much more portable, as they can be folded or rolled up. There are solar panels available from the commercial market that have the potential to provide a higher power output for the same size of panel. A higher power output would supply more power to the equipment in the field, thereby increasing its usefulness.

One possibility for improving the power output of a solar blanket is the use of the silicon-based Sunpower A-300 solar cell (see Figure 1). The design of this cell is different than conventional cells, which give it an advantage with regard to efficiency. In a conventional solar cell, there are electrical contacts on the front and back of the cell. The grid on the front blocks some of the incoming light, which reduces the cell output. The A-300 is built in such a way that the contacts are on the back of the cell; hence, they are called back-surface-contact cells. This design eliminates the grid on top and allows the full surface area of the cell to collect sunlight. The improved design results in efficiencies greater than 20% [2]. The cell is thinner than conventional designs, making it semi-flexible. To provide greater utility for solar blanket applications, the cell must be made thinner in order to improve flexibility. Reducing the thickness of the cell gives less

material for the photons of light to travel through though. The effect is a reduction in the chance of photons to hit and free electrons, which is the basis for solar cell operation. Fewer freed electrons result in a lower power output. Knowing how much the output will drop is important in determining whether a thinner A-300 type cell is a feasible alternative to current solar blankets.

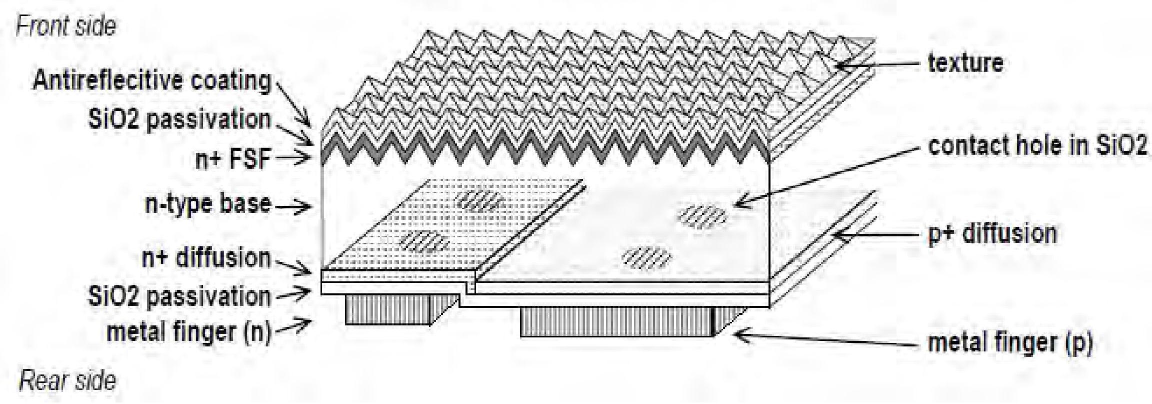


Figure 1. The Sunpower A-300 back-surface-contact solar cell, from [2].

One way to determine the feasibility of a cell is to build and test it. This is very time-consuming and costly. A better alternative is to simulate the device characteristics. Silvaco Atlas software can be used to model semiconductor devices. Solar cells are made of *n* and *p*-type semiconductor material, which can be simulated using Silvaco software. The model parameters in the software can be easily modified and the results observed. This is the method used to investigate the A-300 cell. A model of the cell using the Silvaco Atlas software allowed the thickness to be varied and the results compared to see the effects of reducing the cell thickness.

The A-300 cell modeled in Silvaco was validated by comparing the simulation results with A-300 measurements taken by the National Renewable Energy Laboratory (NREL). The measured values of the A-300 cell are an open circuit voltage (V_{oc}) of 678.0 mV, and a short circuit current (I_{sc}) of 5.885 A. The simulated results were V_{oc}=660.0 mV and I_{sc}=5.92 A. The difference between the real and simulated results shows that the model results are 18.0 mV smaller and 35.0 mA larger than the measured results. Since the two results closely match, this gives a high degree of confidence that the model is accurate and can be used to observe the effects of changing the cell thickness.

The simulation was run and the thickness of the bulk of the cell was varied from 250.0 µm to 5.0 µm. A comparison of the power output between the different thicknesses is shown in Figure 2. As expected, when the thickness is decreased, the output of the cell also decreases. The power output does not decrease significantly until the thickness is reduced to 30.0 µm, which indicates that the cell thickness could be reduced significantly to increase flexibility while maintaining a considerable amount of the power output. Even at the 5.0 µm thickness, the cell power still outperforms that of SPACES. It takes 49 A-300 cells to equal the surface area of SPACES. The output of a 5.0 µm device is 1.93 W per cell, which equates to 94.5 W for an array similar in size to SPACES. This yields a 32.5 W advantage over SPACES.

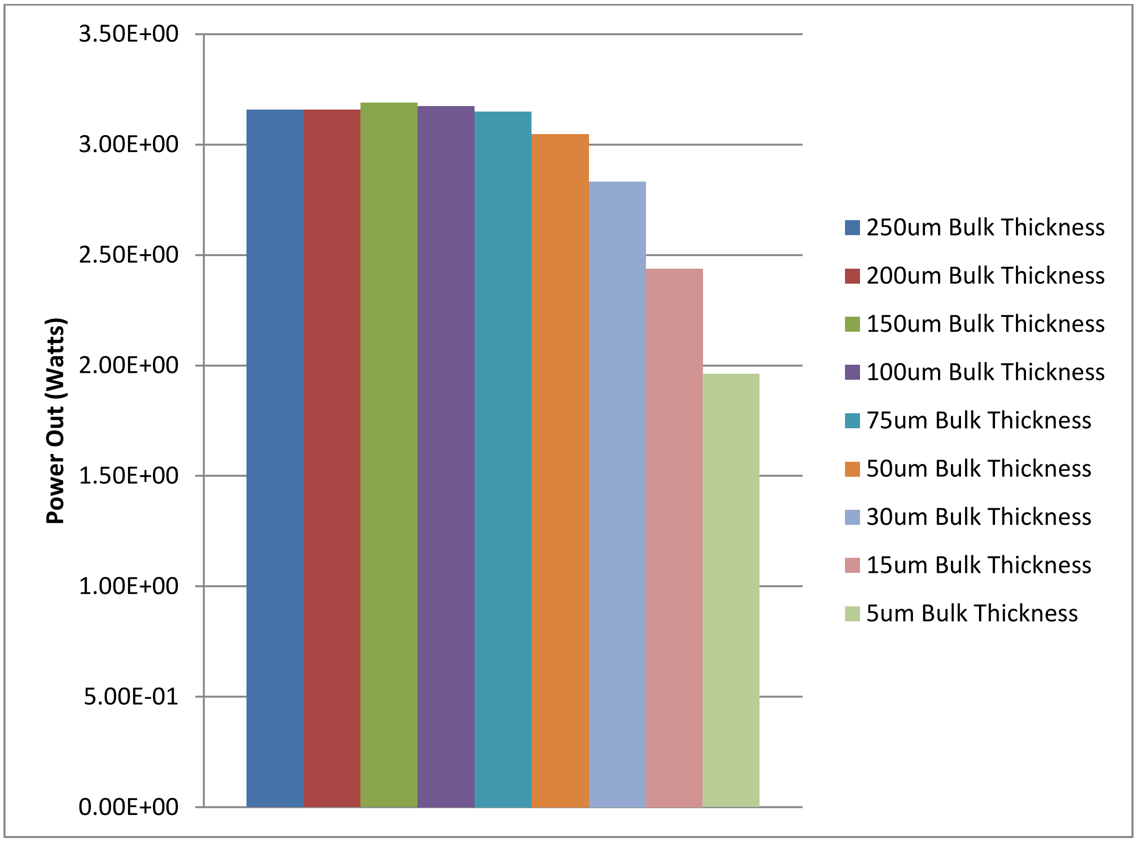


Figure 2. Simulation results of power output for varying bulk thicknesses.

In this thesis, a computer model of the Sunpower A-300 cell was built. Using the model, we ran several simulations to determine the effects of reducing the cell thickness. It was shown that a reduction can be made to the thickness of the A-300 cell while still outperforming the SPACES solar blanket system.

The reduction in thickness in the model was preliminary. The only parameter changed was the bulk thickness. The model has many parameters, and different combinations of those parameters will affect the cell output. Using computer algorithms, we can change the device characteristics and observe the differences in output, allowing for optimization of the cell for a more favorable increase in the power output. Additionally, there are materials other than silicon that are used to manufacture solar cells. These materials can produce a higher output. With this model, alternate materials can be investigated to see whether even more power can be produced from a back-surface-contact solar cell.

LIST OF REFERENCES

[1] Iris Technologies, “SPACES generation II (SPA II) solar portable alternative communications energy system,” Iris Tech., Irvine, CA, Nov. 2013.

[2] W.P. Mulligan, D.H. Rose, M.J. Cudzinovic, D.M. De Ceuster, K.R. McIntosh, D.D. Smith, and R.M. Swanson, “Manufacture of solar cells with 21% efficiency,” Sunpower Corp., Sunnyvale, CA, N.D.

ACKNOWLEDGMENTS

I would like to first thank my wife, Mary Ann, for not divorcing me since I have not been home in the last six months. To my children, Kylie, Brian and Clara, who had to put up with me either being gone or busy all the time. An extra special thanks and hugs go to the littlest one, Henry, who never failed to come running to greet me and hug my legs when I got home after a long day.

To my advisor Professor Michael, thank you for all your guidance and wisdom. Without it, I would have been lost on this seemingly impossible journey that is thesis writing.

Lastly, my gratitude goes out to the Thesis Processing Office for providing such and awesome template and a little training on how to use it. I had no idea MS Word could do so many things. Up until now, I thought it was just typing and spell check.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The use of energy pervades every facet of our daily lives. As technology continues to advance and become integrated in more products, the need for power sources for these devices will only increase. This is equally true of the armed forces, especially when they are deployed to remote areas, which is why the subject of energy is a high priority for the Commandant of the Marine Corps. In 2012, he identified expeditionary energy as a critical area needed for maintaining operational capabilities and readiness.

A. RELEVANCE TO THE DEPARTMENT OF DEFENSE

This research is directly relevant to the needs of the Department of Defense (DOD). In isolated areas it can be difficult to provide resources to ensure daily requirements for forward operating forces are met. In some cases, fuel must be transported great distances overland to reach forward operating bases. The fuel must be protected from attack, which greatly increases cost and puts lives at risk. The ability to reduce fuel demand will positively impact mission readiness. Additionally, when troops deploy from these bases, they are burdened with having to carry batteries to provide power to equipment such as radios when they are out in the field. The batteries add weight and limit the amount of time troops can remain on station before having to recharge the batteries. Frequent charge and discharge cycles increase the need for new batteries, which increases cost and logistical burdens.

To remedy the problems outlined above, systems are available utilizing solar blankets with power electronics that can be used to charge batteries in the field provided there is sufficient sunlight. The Marine Corps currently uses such a system called the Solar Portable Alternative Communications Energy System (SPACES) to provide expeditionary energy to Marines in the field. These systems are relatively lightweight and rugged enough to survive abuse they take in an operational environment. Part of the ruggedness of the solar blanket system comes from the flexibility of the solar panels, which make them less prone to damage. Flexibility also makes the solar cells much more portable as they can be folded or rolled up. There are solar panels available from the commercial market that have the potential to provide a higher power output for the same size of panel. A higher output supplies more power to the equipment in the field. More power would be particularly beneficial in areas with less than optimal sunlight or on a cloudy day where the higher power-producing cells are much better suited to meet electrical demands.

One of the alternatives is a thin film solar cell made out of Cadmium Telluride. The Cadmium Telluride cell is extremely flexible, making it very durable, but there are concerns with the materials of which the cell is made. Telluride is not a very abundant material and could be problematic where large-scale production is concerned. Cadmium is toxic, which creates health and environmental concerns for manufacturing, usage, and end-of-life disposal. Copper Indium Gallium Selenide (CIGS) is another thin film option. These cells are good at absorbing sunlight despite their very thin structure. They have very good efficiencies but cannot best that of the best traditional thick-cell silicon devices. It would be advantageous to have a silicon-based thin cell design that was flexible enough to use in a solar blanket while retaining the efficiencies of the traditional thick-cell devices.

To make a silicon-based cell that is thin enough to have the flexibility needed for a solar blanket application, it must be made much thinner than traditional designs. This limits the cell’s ability to trap light, thereby reducing the output power. Fortunately, there is a design that eliminates the typical grid seen on top of most solar cells. The Sunpower A-300 back-surface-contact solar cell removes the grid, which normally reduces the incoming sunlight to a solar cell by around 8% [1].

B. RESEARCH OBJECTIVE

The feasibility of using the A-300 cell in a military solar blanket application is investigated in this thesis. If the A-300 can be used, it could significantly increase the available power to field-deployed warfighters. A computer model of this type of cell is built using Silvaco Atlas software. The model results are then compared to measured results for an A-300 cell. Once the model is validated, the thickness of the cell is reduced in the simulation to observe the impact on power output. A thinner cell is beneficial for solar blanket applications because it results in a more flexible cell. If the output power remains high for thinner versions of the A-300, this indicates that the cell is a good candidate for military solar blanket applications.

C. PREVIOUS WORK

Prior thesis work has used Silvaco software to model solar cells. This was first done by McCloy [2]. Prior to his work Silvaco had only been used to model transistors, and many in the semiconductor industry were unfamiliar with Silvaco. McCloy was able to use the software to successfully model a single and dual-junction solar cell. His work was continued by Michalopoulos [3], who used Silvaco to model and simulate single-junction Gallium Arsenide, dual-junction Indium Gallium Phosphide, and Gallium Arsenide, and triple-junction cells with Indium Gallium Phosphide, Gallium Arsenide, and Germanium. Work was also done by Fotis [4] on dual-junction CIGS.

The models developed in past theses were all based on conventional solar cell designs. These designs have metal contacts on both the top and bottom of the cells with stacked *n* and *p*-type semiconductor material between. The solar cell model developed in this thesis uses a Back Surface Contact (BSC) cell design. In this type of cell all of the contacts are located on the bottom of the cell, which eliminates the shading usually produced by the top contacts. The effect is an increase in the cell power output. The *n* and *p*-type semiconductor material in this device are placed side by side in alternating layers. The differences in this cell design require building a unique model as compared to previous ones that have been developed.

D. ORGANIZATION

In Chapter II, the power output of SPACES is compared to that of a similar sized A-300 module. In Chapter III, the physics behind solar cells and the BSC type of solar cell design are introduced. The Silvaco atlas simulation software is discussed in Chapter IV, followed by the simulation results in Chapter V. Finally, the thesis conclusions and recommendations for future work that utilize the model developed in this thesis are discussed in Chapter VI. The Silvaco source code used to build and simulate the model is given in the Appendix.

THIS PAGE INTENTIONALLY LEFT BLANK

II. SOLAR BLANKETS

As technology continues to advance, the number of devices we interact with on a daily basis will continue to increase. For the warfighter this means an increase in the amount of energy needed to supply power to those devices. In the field, or even forward deployed, this means significant resources are devoted to ensuring the power is available where it is needed. Having alternative means of supplying power to devices will have a positive impact on overall readiness and warfighting capability.

A. SPACES

The Marine Corps is using a system to provide electrical power to Marines in the field (see Figure 1). The Solar Portable Alternative Communications Energy System consists of a 0.73 m^2 flexible solar blanket, a DC-DC converter, and a battery. The solar panel is rated at 62.0 W, and the system is capable of providing power to operate communications equipment, charge batteries, and other small electronic gear [5]. With this system a Marine in the field does not have to carry as many batteries for things like radios, since the few that are brought can be recharged wherever there is sunlight. The ability to recharge batteries lightens the load that has to be carried and greatly extends the amount of time communications are available.

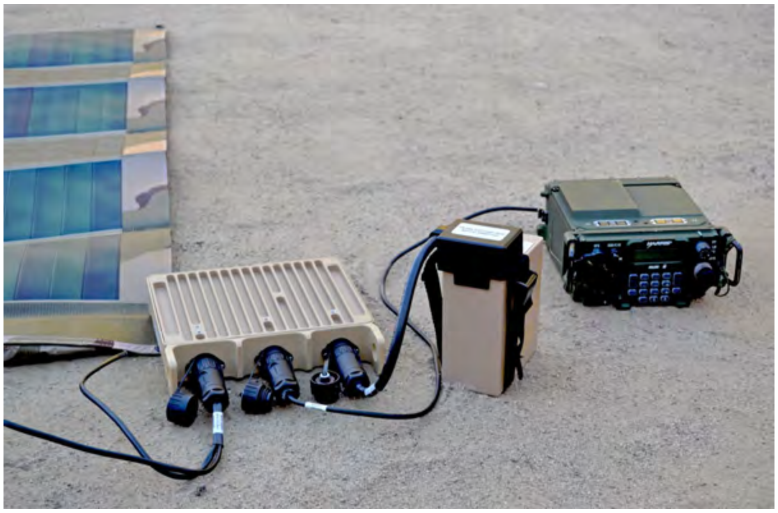


Figure 1. SPACES, from [5].

B. SUNPOWER A-300 CELL

The Sunpower A-300 cell (see Figure 2) is a back-surface-contact solar cell. The A-300 cell is produced using screen-printing technology as opposed to the traditional photolithography used on conventional solar cells. This makes the process faster and more cost effective. The design of the cell eliminates shading on the top of the cell, resulting in one-sun efficiencies measured at 21.5% [1]. This cell is approximately 270.0 µm thick, making it somewhat flexible [6]. Flexibility is one of the characteristics needed when considering cells for a solar blanket. If the cell were thinner, it would be more flexible but would lose some output power since the thinner cell would not be able to absorb as much sunlight. Any solar blanket has to be rugged enough to withstand the abuse it is likely to encounter in the field.

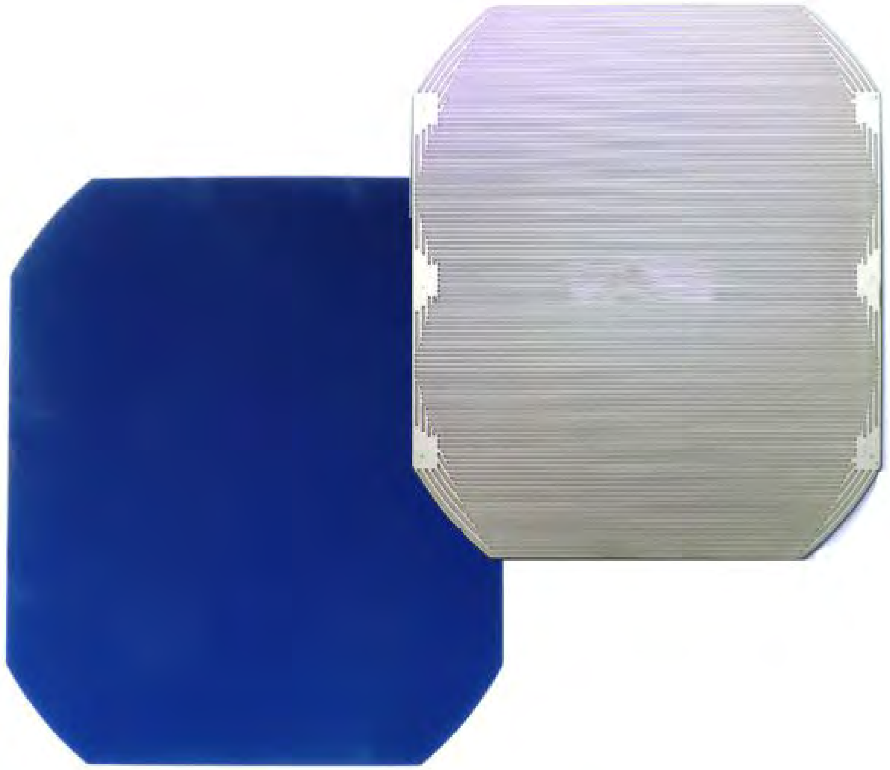


Figure 2. Sunpower corporation A-300 back-surface-contact solar cell, from, EVTV Motor Verks, accessed 15 April 2015, http://store.evtv.me/proddetail.php?prod=sunpower.

C. SPACES VS. AN A-300 BLANKET

SPACES was tested in Monterey, California, in October 2012. The data obtained showed that the system consistently delivered less than the stated 62.0 W of output power (see Figure 3).

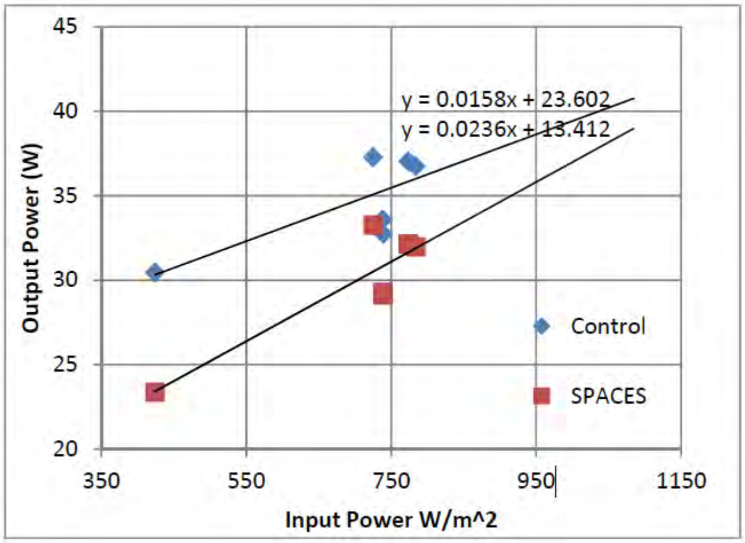
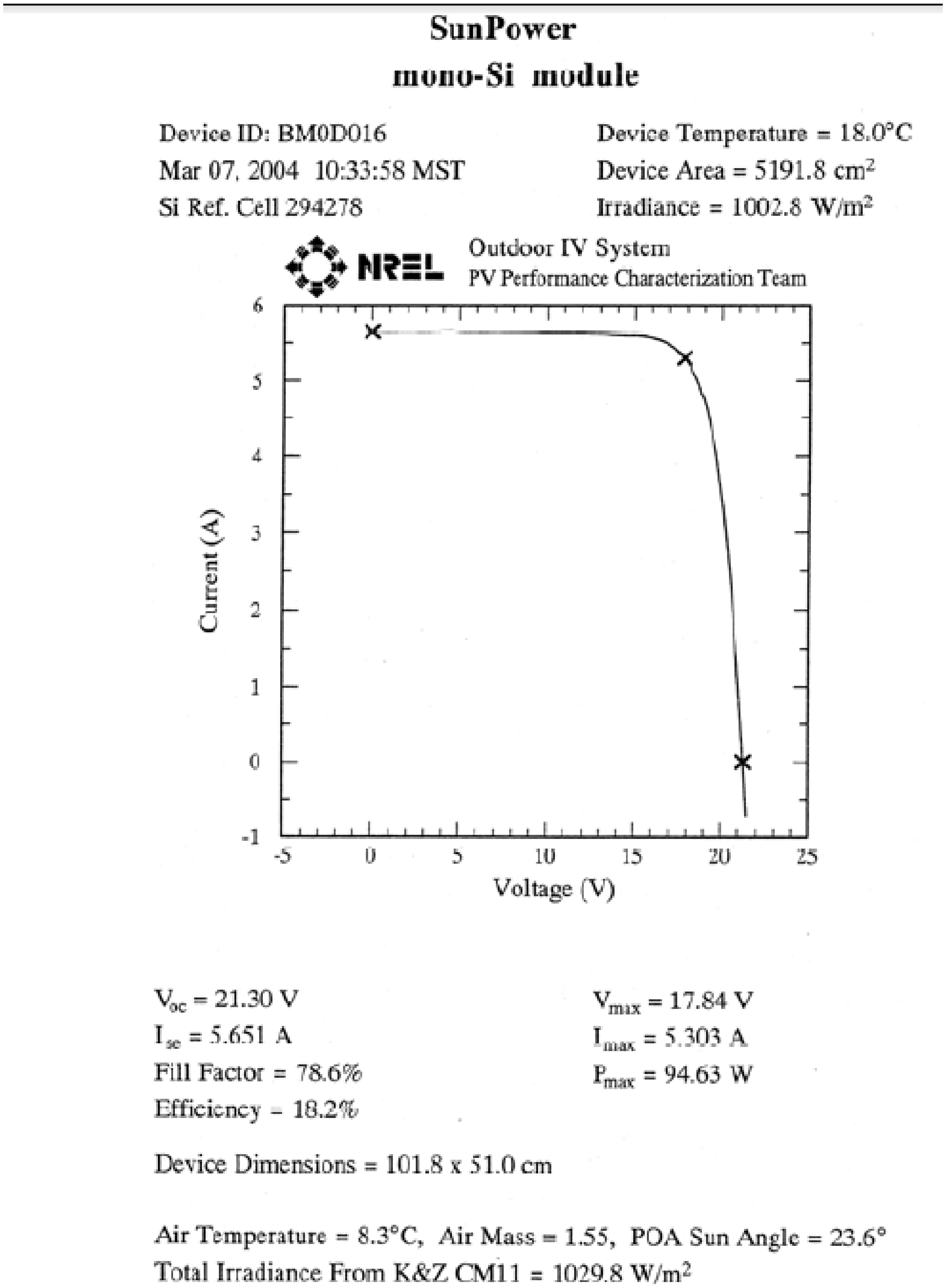


Figure 3. SPACES data taken by research performed at NPS, from [7].

Similar tests were conducted on the A-300 cell by the National Renewable Energy Laboratory (NREL) using a solar irradiance of 1029.8 W/m^2. To compare the two cells at the same input power level, a plot of the measured output power vs. input power (solar irradiance) of SPACES shows a best fit line of

$$y = 0.0236x + 13.412. \tag{2.1}$$

For an input of 1029.8 W/m^2, the expected output power is 37.72 W. The NREL test used a module of 32 A-300 cells. Each cell is 0.0149 m^2, which gives a total area of 0.4768 m^2. At the maximum power point, the output power was measured at 94.63 W (see Figure 4). This yields 2.96 W per cell. Compared to SPACES, the A-300 configuration is smaller yet delivers significantly more power. Even considering the manufacturer's stated 62.0 Watt output, the A-300 exceeds this by over 30.0 W. It would take 49 A-300 cells to equal the same area as SPACES. If the size of the A-300 were increased to match that of SPACES, the output would approach 145.0 W, more than twice the best possible SPACES output.



SunPower
mono-Si module

Device ID: BM0D016
Mar 07, 2004 10:33:58 MST
Si Ref. Cell 294278

Device Temperature = 18.0°C
Device Area = 5191.8 cm^2
Irradiance = 1002.8 W/m^2

NREL Outdoor IV System
PV Performance Characterization Team

V_{oc} = 21.30 V
I_{sc} = 5.651 A
Fill Factor = 78.6%
Efficiency = 18.2%

V_{max} = 17.84 V
I_{max} = 5.303 A
P_{max} = 94.63 W

Device Dimensions = 101.8 x 51.0 cm

Air Temperature = 8.3°C, Air Mass = 1.55, POA Sun Angle = 23.6°
Total Irradiance From K&Z CM11 = 1029.8 W/m^2

Figure 4. NREL test of a 32-cell A-300 solar module, from [1].

III. SOLAR CELL PHYSICS

Solar cells are made from semiconductor material. Semiconductors have properties somewhere between that of insulators and conductors. Electrical conductors are materials in which some of the electrons are free electrons that are not tightly bound to atoms. These electrons can move relatively freely within the material. Electrical insulators are the opposite. Most of the electrons in an insulator are bound to atoms and cannot move easily within the material[8].. The number of electrons available for conduction in semiconductors can be greatly increased by the addition of thermal or optical energy [9]. When sunlight hits this material, it generates electron hole pairs, which then can be collected and used to generate electricity. The most common semiconductor material used in solar cells is silicon. It is abundant, and its material properties have been extensively studied.

A. DOPING OF SEMICONDUCTORS AND P-N JUNCTIONS

If a silicon atom is replaced with another type of atom, such as a group III material that only has three valence electrons, there will be a silicon atom that does not have four electrons it can share. The vacancy results in a hole that easily accepts a free electron. If more silicon atoms are replaced, there will be more holes. The resulting material is called *p*-type (see Figure 5).

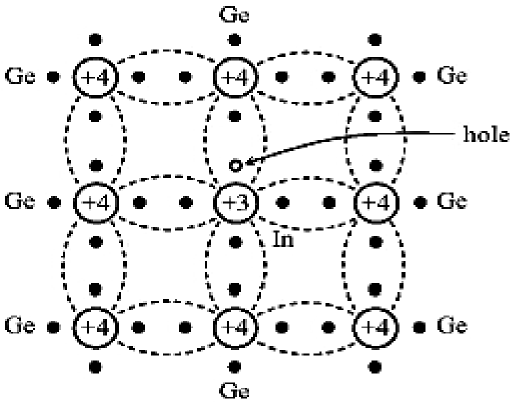


Figure 5. *p*-type material created by replacing a Germanium atom with an Indium atom, after [10, p. 17].

Given this analogy, if the silicon atom was instead replaced with a group V material that has five electrons in its outermost shell, there is instead an extra electron. Since this electron is not tightly bound, it is free to move throughout the material, making it *n*-type (see Figure 6).

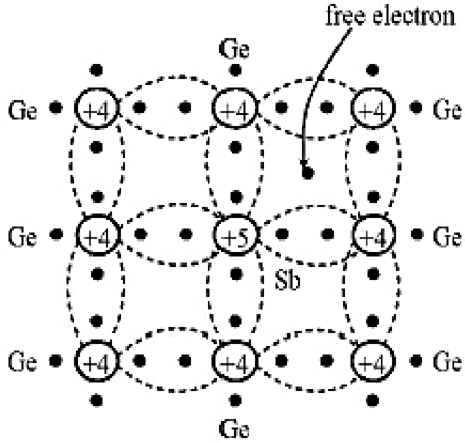


Figure 6. *n*-type material created by replacing a Germanium atom with an Antimony atom, after [10, p. 15].

When a *p*-type and *n*-type semiconductor material are joined to each other they form a junction. The excess electrons from the *n*-type material and the excess holes from the *p*-type material are attracted to each other and combine in the vicinity of the junction. This area lacks any extra holes or electrons and is called the depletion region. Within the depletion region, the area on the *n*-type side loses electrons that diffuse to the *p*-type side and fill in the holes. Prior to the movement of electrons, the *n*-type side is electrically neutral. The loss of electrons gives the *n*-type side (within the depletion region) a positive net charge. The *p*-type side also starts out as electrically neutral. With the extra electrons, it now has a negative net charge. The presence of positive and negative areas creates a voltage difference between the *n* and *p* regions of the semiconductor material, which is the barrier voltage V_o, which must be overcome for current to flow through the device (see Figure 7).

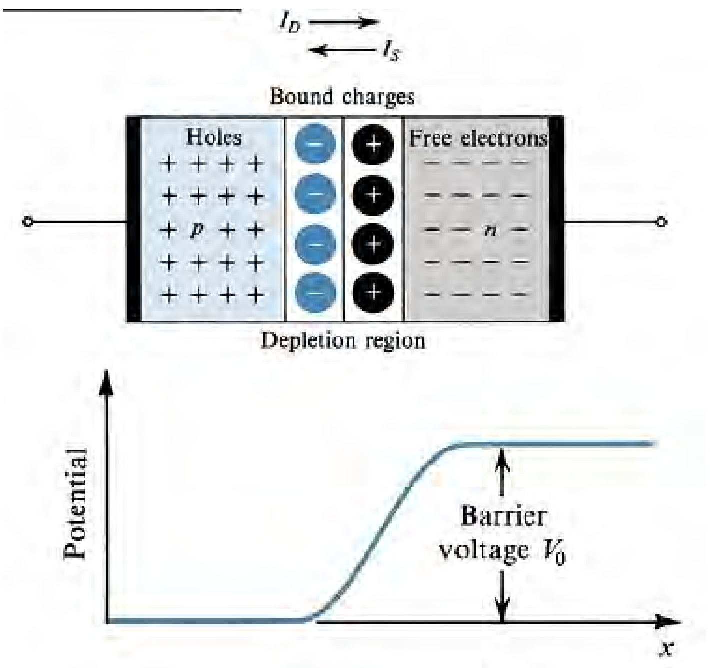


Figure 7. The barrier voltage V_o as a result of the PN junction, after [10, p. 21].

B. SOLAR CELLS

The PN junction is the basis for the construction of a solar cell. When the *p* and *n* semiconductor materials are laid flat on top of each other and exposed to sunlight, the possibility exists that a photon will hit an electron and transfer enough energy to free it. Connecting metal contacts to the *n* and *p* materials gives a path for the electrons to flow. An electrical load can be connected via these conductors to utilize the generated electrical energy (see Figure 8).

1. Energy Bands

An electron residing in its atom’s valence shell is referred to as being in the valence band. If the electron is free it is considered to be in the conduction band. The amount of energy it takes to move the electron from the valence band to the conduction band is the bandgap. A diagram showing the energy bands of different materials (see Figure 9) shows that the bandgap of insulators is very wide, meaning it takes a significant amount of energy to free one of its electrons. The bandgap of conductors is non-existent because the conduction and valence bands overlap. This overlap explains why electrons flow easily in a conductor. The semiconductor has a bandgap between that of an insulator and conductor. In a solar cell, the narrow bandgap of the semiconductor material is what allows electrons to be freed. When a photon passes through a semiconductor and hits an electron, it gives up energy to the electron. The shorter the wavelength of the photon, the more energy it has to give to the electron. Any photons with an energy higher than the bandgap of the material can move an electron to the conduction band, thereby generating an electron-hole pair.

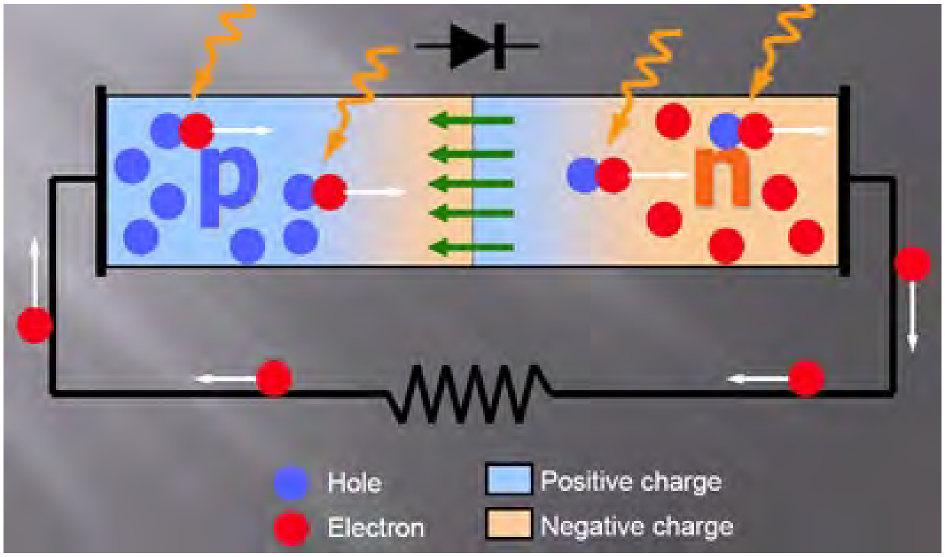


Figure 8. Solar cell with external connected load, from [10, p. 47].

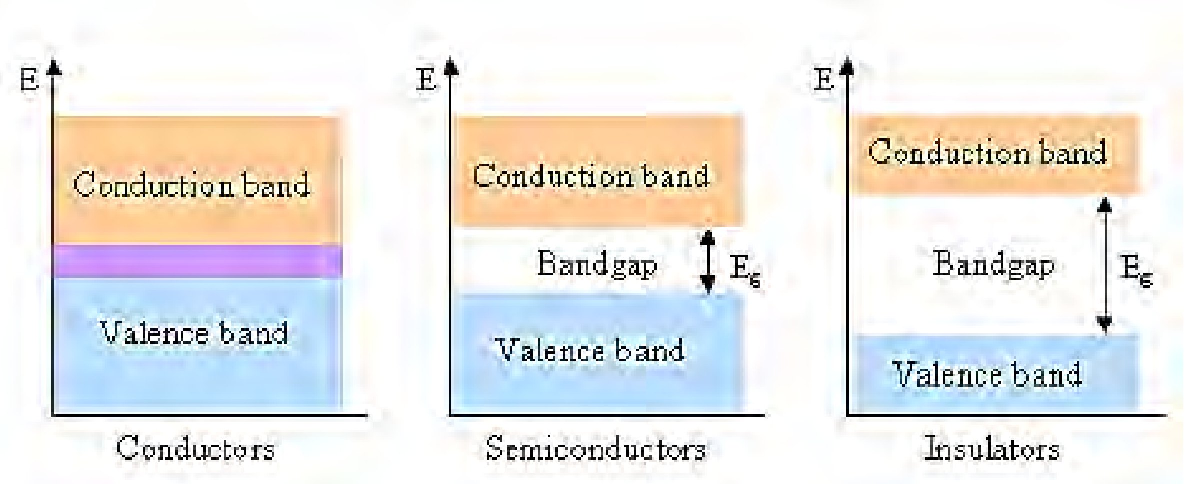


Figure 9. Bandgaps of conductors, semiconductors, and insulators, from [11,

p. 9].

2. Solar Spectrum

Sunlight is made up of electromagnetic waves, with wavelengths from 0.3 microns up to about 2.5 microns. The intensity of solar energy is highest around the visible wavelengths and decreases to a negligible amount in the infrared region. The solar spectrum outside of the earth's atmosphere is termed the AM0 spectrum. The surface spectrum is termed AM1. A plot of solar intensity on the earth's surface shows drops in energy at several wavelengths as shown in Figure 10. The drop is because, as the solar radiation travels through the earth's atmosphere, some of the energy is absorbed. The wavelengths that have a drop in intensity are the ones that are more easily absorbed due to the composition of gases that make up the atmosphere.

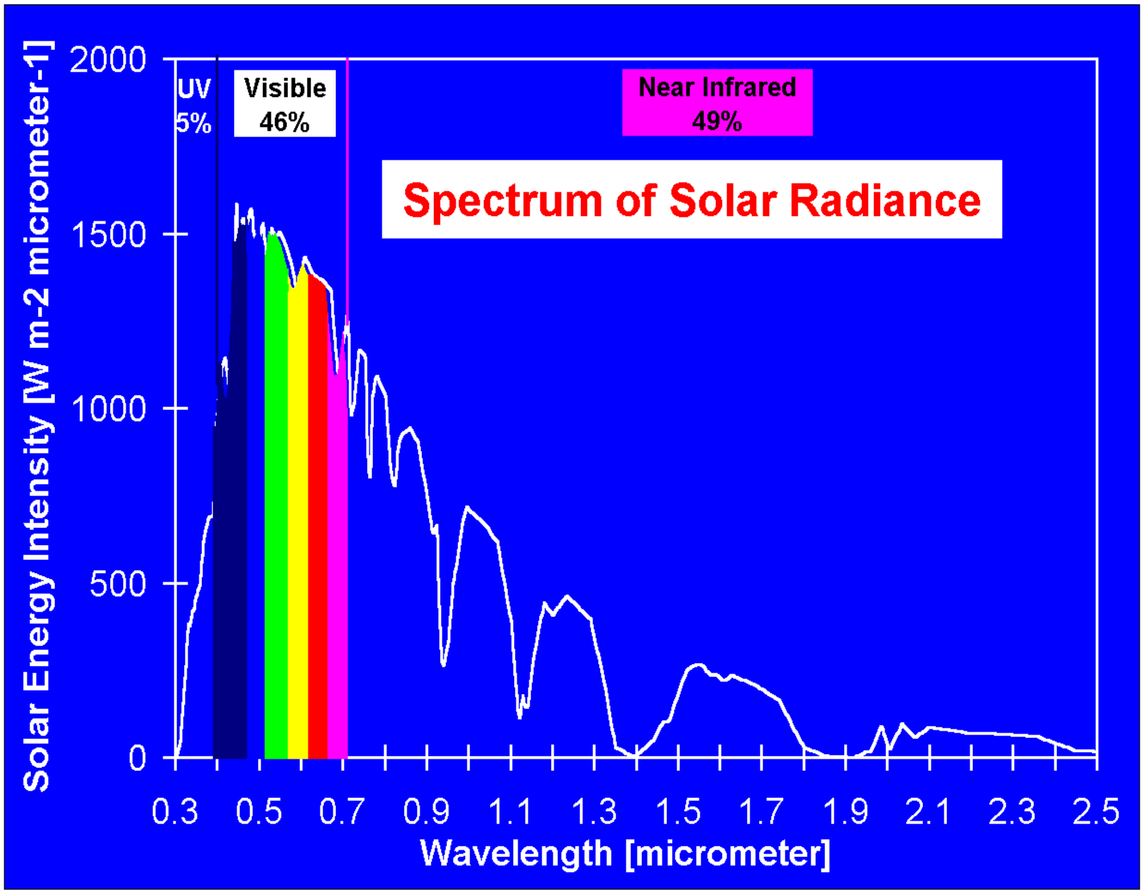


Figure 10. Spectrum of solar radiance, from [12].

3. PV Cell Efficiency

There are many factors that affect the efficiency of a solar cell. Metal contacts are necessary to connect the semiconductor material to an external load. Normally, connections at the top of a cell are arranged in a grid pattern. These connections allow connection to a load while also allowing sunlight to reach the semiconductor material; however, the areas covered by the grid reflect sunlight. The result is a reduction in the surface area that can collect light, which is referred to as shading. Shading reduces the efficiency of a cell by 8% to 10%. The grid also introduces losses due to the resistance of the grid wires. The grid can be made thicker to reduce power losses, but this increases shading. Conversely, making the contacts thinner to reduce shading increases the grid resistance, which increases the resistive losses. This becomes an optimization problem to balance these two loss mechanisms so that maximum power can be achieved. There are also resistive losses in the semiconductor material and at the contact junctions. The grid is not the only part of the solar cell that can reflect the sunlight. The semiconductor material also reflects some of the incoming light. To combat this, an antireflective coating is used. Another method is to texture the surface to change the angle that the light hits the surface of the cell. If the light is reflected, there is a chance it will reflect towards an adjacent surface where it can be absorbed. This geometric effect is shown at the top of Figure 11.

Even if the photon makes it into the material, there is a chance that it will not free an electron. In order to pass its energy to an electron, the photon must hit it. On this small scale, there is a relatively large amount of room between atoms, and the photon can go all the way through the material and not contact an electron. Back side reflectors can reflect the photon back up through the material giving it another chance to free an electron. There is still the possibility that the photon will pass through the material and exit out of the top. When a photon does succeed in freeing an electron, there is a hole left in the spot where the electron was. The generated electron hole pair needs to separate and travel to their respective electrical contacts. These pairs are generated at different depths within the semiconductor material. As they make their way to the contacts, they encounter other holes and electrons, opening up the possibility that they will recombine and be lost (see

Figure 12). This recombination is yet another loss mechanism in the solar cell. Imperfect crystal structures and impurities in the semiconductor material increase the recombination losses. Using a material with a high carrier lifetime, high purity, or minimizing the thickness are ways to reduce the recombination losses. Care must be taken as reducing the thickness reduces the chance that a photon will generate an electron-hole pair.

Finally, if a photon encounters an electron, there is still no guarantee that the electron will be freed because it takes a certain amount of energy to move the electron into the conduction band. The bandgap for silicon is 1.11 eV [9]. The relationship of the bandgap energy to wavelength is

$$E_g = \frac{hc}{\lambda} \tag{2.2}$$

where E_g is the bandgap energy, h is Planks' constant, c is the speed of light, and λ is the wavelength. The required wavelength to free an electron in silicon is 1.12 microns, which is in the infrared spectrum. Solar cells operate over a much larger spectrum. Any wavelength larger than 1.12 microns will not have enough energy to free an electron. If the wavelength is shorter than 1.12 microns, it will have more than the required 1.1 eV needed to free an electron. The electron will be freed and move past the conduction band. It will then give up the extra energy as heat and move back down to the conduction band. In this case, heat is transferred to the solar cell. The lattice vibrations from the excess heat reduce efficiency by interfering with the flow of charges. Regardless of the heat produced, the higher energy photon still generates an electron-hole pair. Use of higher bandgap materials reduces the amount of photons that exceed the bandgap energy required to free an electron and more efficiently targets high energy photons for electron-hole pair generation. Using different combinations of materials and placing them in layers creates multiple junctions that takes advantage of the different band energies. These combinations help to maximize the efficiency of the cell.

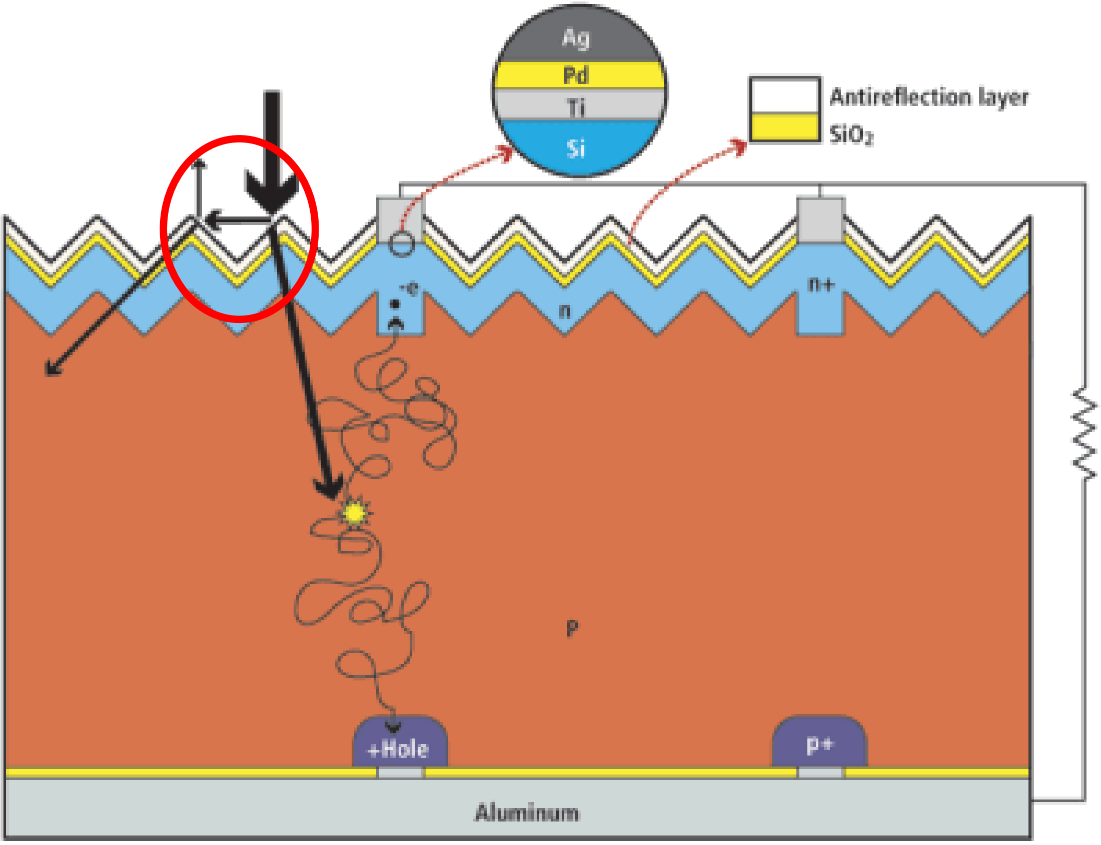


Figure 11. Solar cell with an anti-reflective coating and texturing to maximize light capture, from [13].

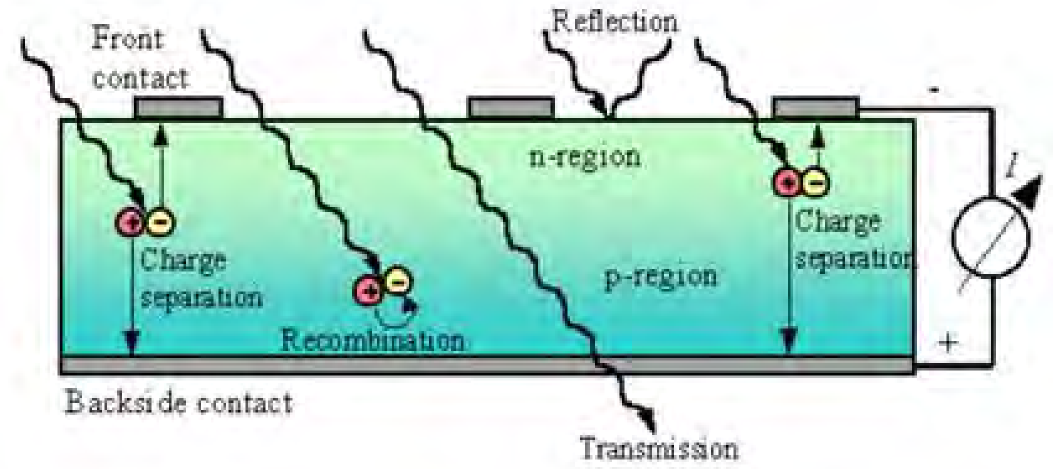


Figure 12. Solar cell with a generated electron hole pair recombining, from [10].

4. Solar Cell Parameters

There are several parameters of interest when discussing solar cells and their output. A typical IV curve for a solar cell is shown in Figure 13. The open circuit voltage (V_{oc}) is the point on the curve where the current is zero. The short circuit current (I_{sc}) is where the voltage is zero. These two values show the maximum voltage and current output of the cell and are a good cursory indicator of performance. A better indicator is the maximum power (P_{max}) and fill factor (FF). The maximum power point (MPP) is the point on the IV curve with the highest output power (see Figure 14).

An important consideration with solar cells is to get the highest power output possible. If a cell is operated at a lower or higher voltage than the P_{max} voltage, there is power lost that could have been used. The FF gives an indication of how square the IV curve is. The faster the IV curve drops off, the squarer it is, and a larger P_{max} can be achieved. An easy way to understand this is that the area under the IV curve is power. At any point on the curve, a rectangular box can be drawn to the current and voltage axes. This rectangle represents the amount of usable power. Anything under the curve not encompassed by the rectangle is lost power (see Figure 15). If the IV curve were square, then the rectangle would contain all the area under the IV curve. When this is the case, all of the power is utilized. Fill factor is given by

$$FF = \frac{P_{\max}}{V_{oc} I_{sc}} 100\%. \tag{2.3}$$

The last important parameter is the efficiency (η), which is the best indicator of performance because it relates the power input to the power output. The higher the efficiency, the better the cell. To calculate η, the solar irradiance must be known. Irradiance can be found using a light measuring device. If no device is available, there are look up tables that list the average irradiance for various locations. Once this is known, multiplying the intensity by the area of the cell gives the power input (P_{in}). The power output is simply P_{max}. Dividing the two yields

$$\eta = \frac{P_{\max}}{P_{in}} 100\%. \tag{2.4}$$

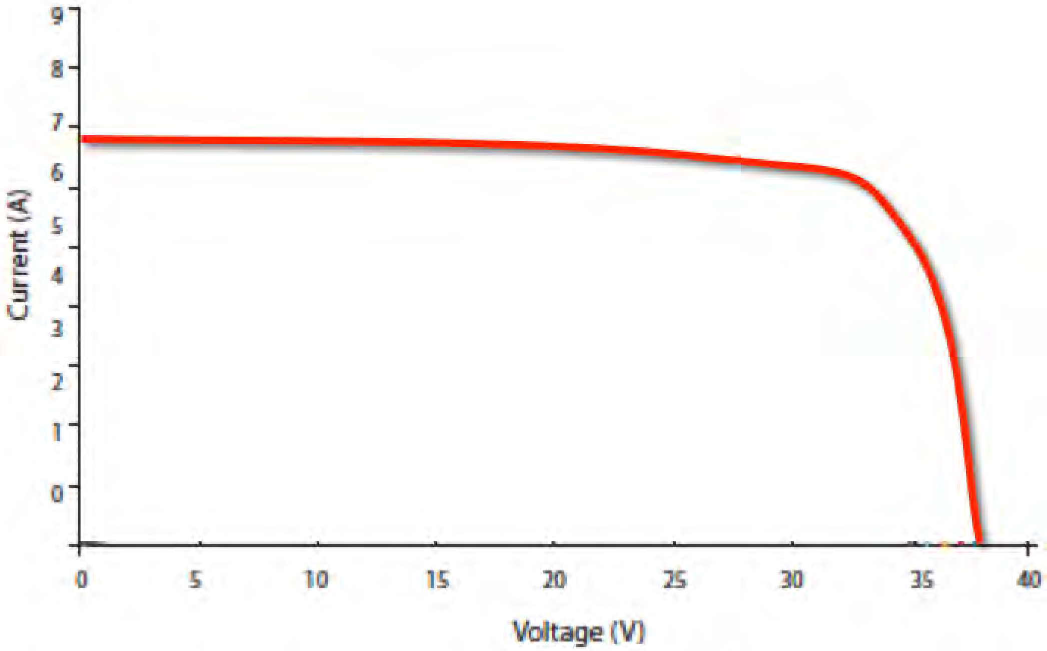


Figure 13. Typical IV curve for a solar cell, from [14].

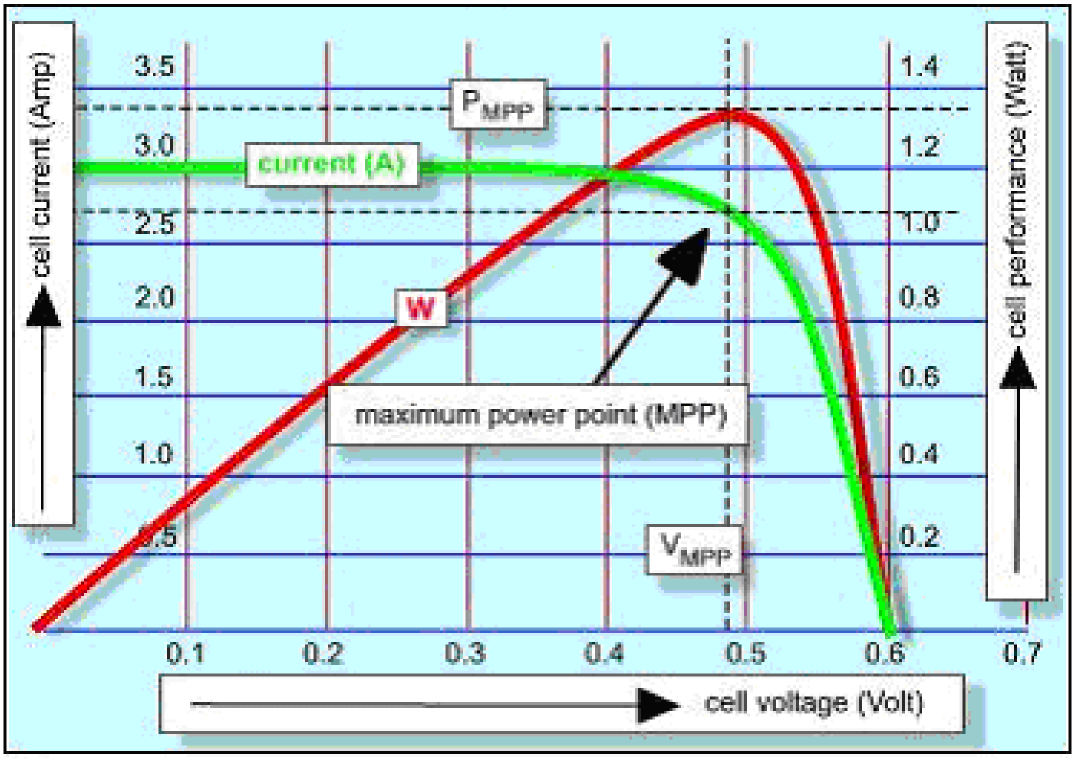


Figure 14. IV curve for a solar cell showing the MPP. The line is red represents the output power at various points on the IV curve, from [15].

When comparing solar cells, it is important to ensure that they are compared for similar operating conditions. Since temperature affects cell efficiency, two cells tested at different temperatures is not a good comparison. Likewise, if two cells are tested using different spectrum or solar concentrations, it is difficult to tell which cell is actually better since the test conditions affect the output.

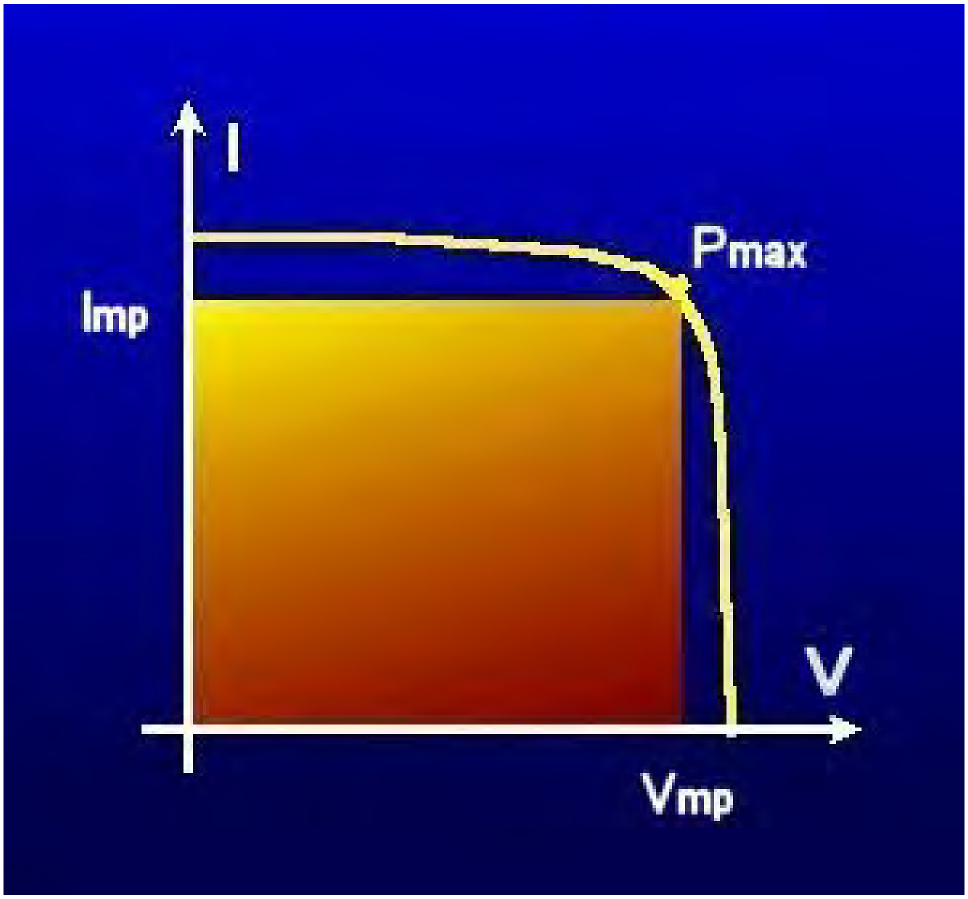


Figure 15. IV curve with the MPP. The orange area is usable power, the two blue areas between the IV curve and orange area is the lost power, from [16].

C. INTERDIGITATED BACK-SURFACE-CONTACT SOLAR CELLS

The BSC solar cell in Figure 16 operates on the same principles as traditional solar cells. The design is significantly different though, which offers advantages over conventional designs in increased efficiency, power output, and spectral response.

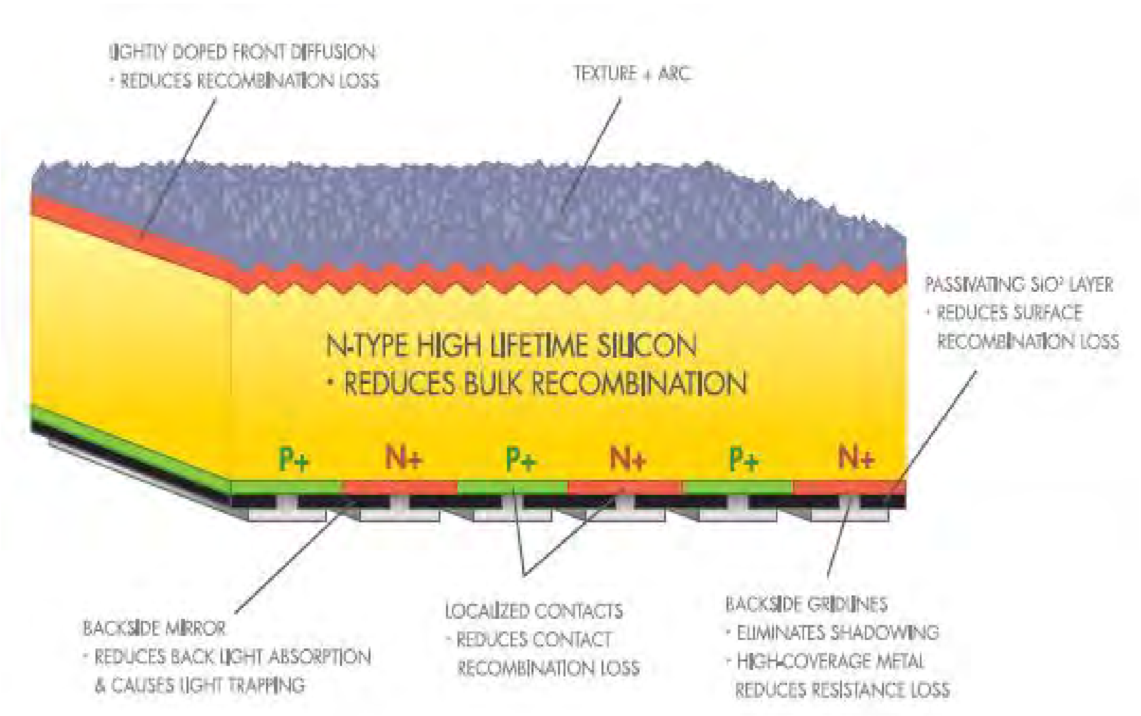


Figure 16. Back surface contact solar cell diagram, from [17].

1. Contacts

The primary difference between the BSC cell and the more common designs is the lack of a grid on top of the cell. Standard solar cells have *n*-type material stacked on top of a *p*-type semiconductor material, with a metallic grid on top and a metallic coating on the bottom. These contacts serve as the electrodes that collect the carriers generated within the cell. The design of the BSC cell has the *p* and *n* material placed side by side on the bottom, with the electrodes connected to them in an interdigitated pattern (see Figure 17). With the electrodes on the bottom, there is no need for the grid on the top of the cell. The lack of a grid eliminates losses due to shading of the semiconductor material, which increases the amount of captured sunlight by eight to ten percent. As explained in Chapter III.3, the shading can be minimized for conventional cells by decreasing the wire size at the cost of increasing resistive losses. Since the connections in the BSC cell are on the bottom and shading is not a factor, the grid wires can be made much bigger, which reduces the series resistance and lowers resistive losses [18].

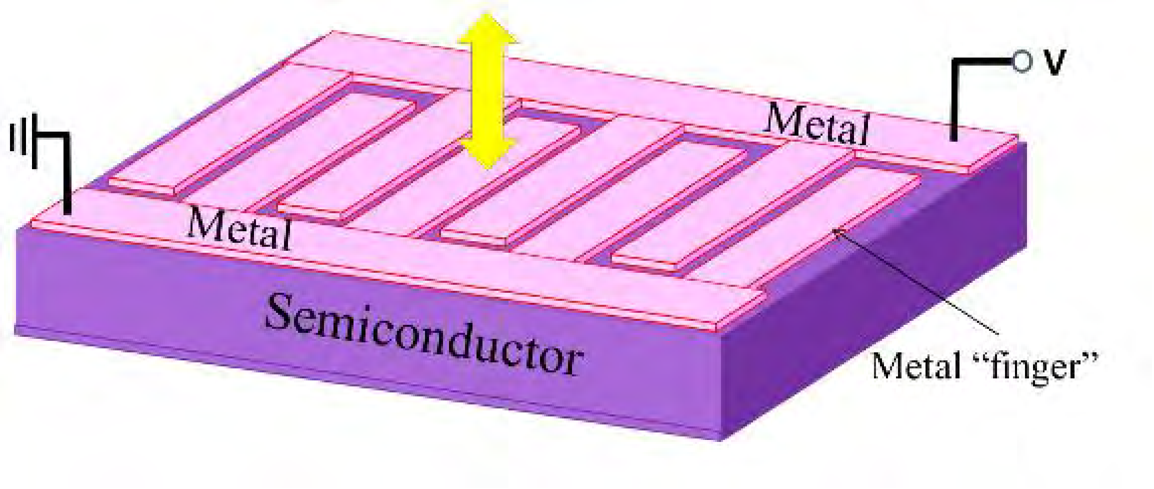


Figure 17. Representation of the bottom of a BSC cell showing the interdigitated pattern, from [19, p. 94].

2. Top Layers

Freeing the top of the cell from the grid simplifies the manufacturing process and allows the top of the cell to be textured. Texturing helps reduce the amount of reflected light and allows stationary solar panels to capture more sunlight as the angle between the sun and panel changes (see Figure 18). To further improve the light capturing capability of the cell, an anti-reflective (AR) coating is placed on top of the texture. A non-coated surface can reflect up to 30% of the incoming light. Adding the AR coating reduces this to about 10% [8]. The combination of the texturing and AR coating can reduce the total reflection down to five percent or less. [10].

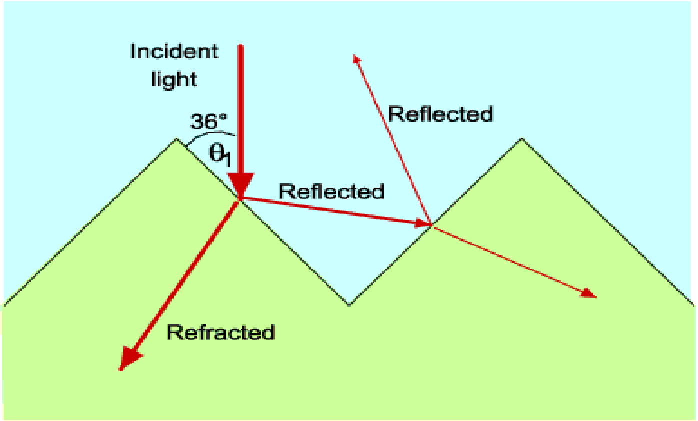


Figure 18. Textured cell that helps capture more light, from [20].

The layer underneath the AR coating is a passivation layer usually consisting of silicon dioxide intended to reduces the number of defects at the silicon interface, which lowers the surface recombination velocities [21, p. 14]. A low surface recombination velocity is a key parameter for the solar cell to be efficient. The next layer is heavily doped with an acceptor material to make it *n*+ in order to create an electric field referred to as the Front Surface Field (FSF), which reduces the number of minority carriers to further lower surface recombination velocities. A recombination velocity below 10.0 cm/s is needed to achieve efficiencies near 22% [21, p. 21]. An optimized FSF also improves UV-stability, low illumination performance, and reduces the majority carrier lateral series resistance [22, p. 68–69].

3. Middle Layer

Another important parameter to achieving higher efficiencies is a long carrier lifetime. The minority carriers must travel through the entire thickness of the wafer to reach the electrodes on the bottom of the cell. Without a high lifetime, the carriers will recombine and the solar cell will have little to no output. To lower the chance of recombination, the carrier lifetime must be greater than one millisecond [1]. The majority of the thickness resides in the bulk layer of the solar cell. This bulk area is lightly doped *n*-type material, and its main job is to capture the light for photo-generation. If the bulk is too thin, then there is less of a chance for an electron-hole pair to be generated. Conversely, if the bulk is too thick, there is a greater chance for recombination.

4. Bottom Layers

Underneath the bulk layer are the heavily doped *n* and *p* regions. The doping concentration is around 10^{20} atoms/cm^3, and these regions function to capture the generated carriers. The *p* region is made larger than the *n* region and creates a shorter path for the minority carriers, which are the holes (see Figure 19). Since holes move more slowly than electrons, they are more likely to recombine before they traverse the cell. The shorter path reduces these recombination losses. The electrons, on the other hand, move much faster and are less likely to recombine.

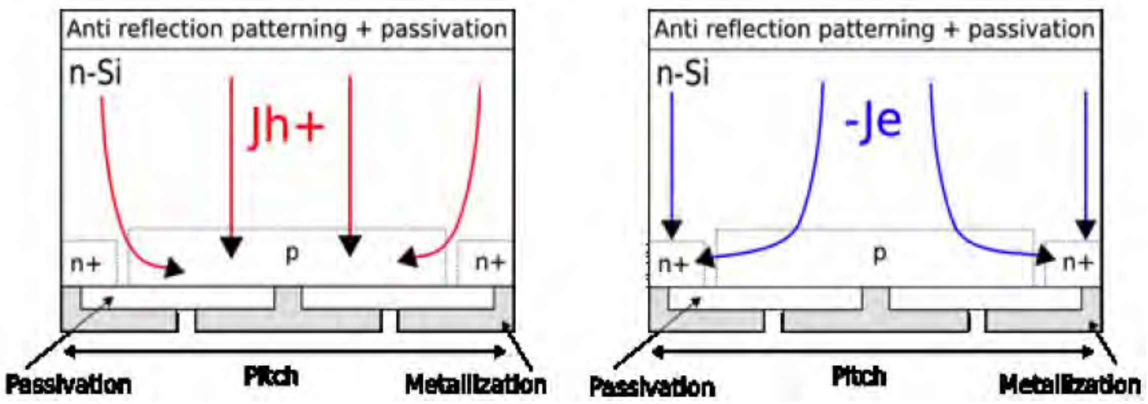


Figure 19. Pathways the minority carriers take to reach the electrodes, from [21].

Below the *n* and *p* regions is a passivation layer consisting of SiO2 that improves the long wavelength response of the cell by reflecting the longer waves that were not absorbed back through to the top of the cell [1]. This layer gives the waves another opportunity to create an electron-hole pair. It also has pathways through to the *n* and *p* layers to connect the *n* and *p* regions to the electrodes on the bottom of the cell. The openings are made small to reduce recombination losses at the interface between the metal contacts and the semiconductor material. [21]. The small openings also give a larger area for the SiO2, which increases the amount of light reflected back towards the top of the cell.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. SILVACO ATLAS SIMULATION SOFTWARE

Silvaco Atlas is a software package used to simulate semiconductor devices. It predicts the electrical behavior of a device, which can be modeled in either two dimensions (2D) or three dimensions (3D). The software consists of several integrated programs that work together to achieve the desired results. The main programs are Deckbuild, Devedit, Tonyplot, and Athena, but there are several subprograms that are accessed during simulation that serve more specific functions. These programs give the user the ability to simulate the production process to manufacture a semiconductor device and test its characteristics. There are many models, numerical methods, and types of material built into the program, giving a wide range of functionality to the user. This allows the modeling of anything from simple devices to complex circuits [23].

A. DEVICE MODELING

Atlas is a physically-based device simulator that predicts the electrical characteristics based on physical structures and bias conditions. It does this by applying a set of differential equations based on Maxwell's laws and the semiconductor transport equations to the nodes of a grid overlaid on the device. This allows the simulation of transport carriers through the structure.

1. Semiconductor Equations

The three equations derived from Maxwell's laws are Poisson's Equation, the continuity equations, and the transport equations. Poisson's Equations relates the electrostatic potential to the space charge density and is given by

$$div\left(\varepsilon\nabla\Psi\right) = -\rho. \tag{4.1}$$

where ε is the local permittivity, Ψ is the electrostatic potential, and ρ is the local space charge. The carrier continuity equations for electrons and holes are defined, respectively, by

$$\frac{\partial n}{\partial t} = -\frac{1}{q} div \vec{J}_n + G_n - R_n. \tag{4.2}$$

and

$$\frac{\partial p}{\partial t} = -\frac{1}{q} div \vec{J}_p + G_p - R_p. \tag{4.3}$$

where n and p are the electron and hole concentrations, $\vec{J}_n$ and $\vec{J}_p$ are the electron and hole current densities, G_n and G_p are the generation rates for the electrons and holes, R_n and R_p are the recombination rates for electrons and holes, and q is the charge of an electron. The transport equations come from the Boltzmann Transport Equation. Applying approximations and simplifications to this equation results in the drift-diffusion model, the energy-balance transport model, or the hydrodynamic model. The drift-diffusion model is the simplest model but is less accurate for devices with smaller feature sizes. When small devices are used, it is then necessary to use the energy-balance or hydrodynamic models.

The drift-diffusion model is used in this thesis to model the current density. For this model,

$$\vec{J}_n = qn\mu_n \vec{E}_n + qD_n \nabla n. \tag{4.4}$$

and

$$\vec{J}_p = qp\mu_p \vec{E}_p - qD_p \nabla p. \tag{4.5}$$

where μ_n and μ_p are the electron and hole mobilities, $\vec{E}_n$ and $\vec{E}_p$ are the electric field strengths, n is the electron concentration in the conduction band, p is the hole concentration in the valence band, D_n and D_p are the diffusion coefficients for electrons and holes, ∇n and ∇p are the electron and hole concentration gradients. Equations (4.4) and (4.5) account for electron and hole current densities resulting from drift and diffusion of carriers, respectively. Adding an electron generation term to the right side of equation (4.4) and a hole generation term to the right side of equation (4.5), we get a model that accounts for current generated from a light source. It is in this way, Silvaco is used to simulate a solar cell.

2. Numerical Methods

To simulate the device, Atlas may have to solve up to six equations depending on the models selected. For each model type there are three techniques, decoupled Gummel, fully-coupled Newton, and Block. The Gummel method solves each unknown while keeping the other variables constant. It continues to do this until a stable solution is found. The Newton method solves the total system of equations together to find a solution. The Block method solves the equations using a combination of the Newton and Gummel methods, in that some equations are solved fully coupled and the rest are de-coupled.

B. DECKBUILD

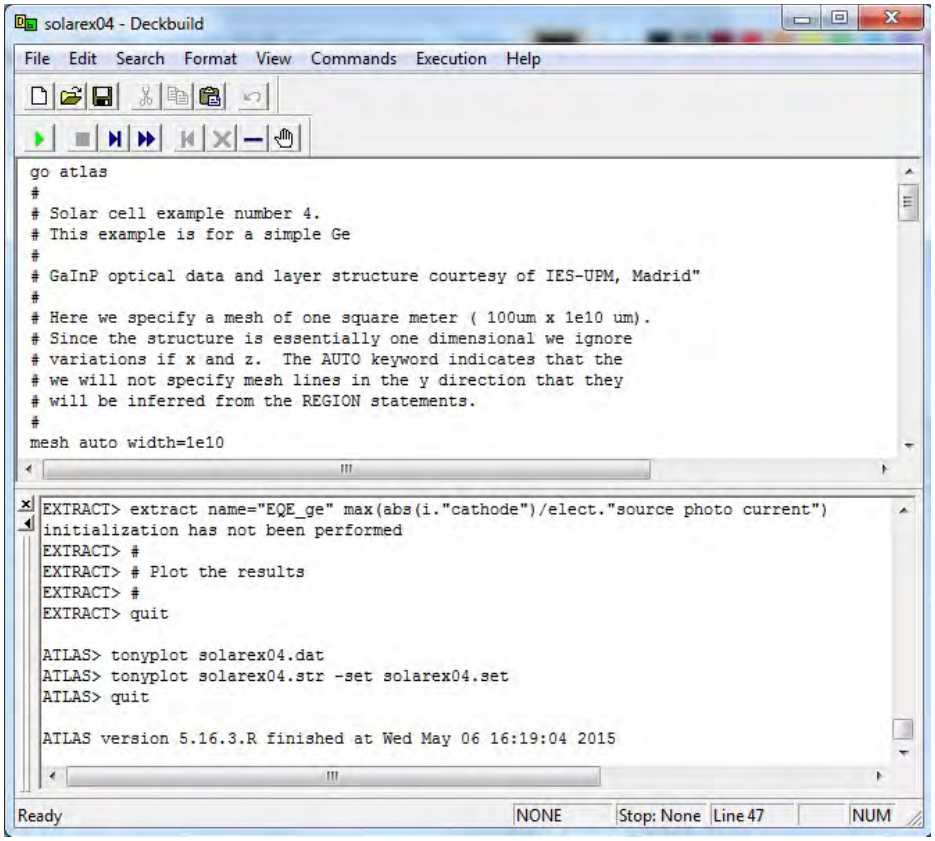
Deckbuild (see Figure 20) is the main program that runs the simulation and calls the associated programs as needed. Deckbuild uses command line code to designate what and how to run. To run Atlas simply type:

```
Go atlas
```

This runs the Atlas program within Deckbuild and is usually the first command unless running one of the other programs, such as Athena, first. Once Atlas is initiated, the next step is to set the parameters of the device. Atlas has a specific order (see Figure 21) in which the statements must be placed, otherwise the program may not function correctly. Even if it does run, it is possible that certain parameters may not be used, which causes inaccurate results. Generally the format is:

```
<STATEMENT> <PARAMETER>=<VALUE>
```

Statements can have more than one parameter defined. Deckbuild provides many built-in examples of different devices that allow a user to run the simulation and view the results. The examples are helpful to see how the code is written when unsure how it is used.



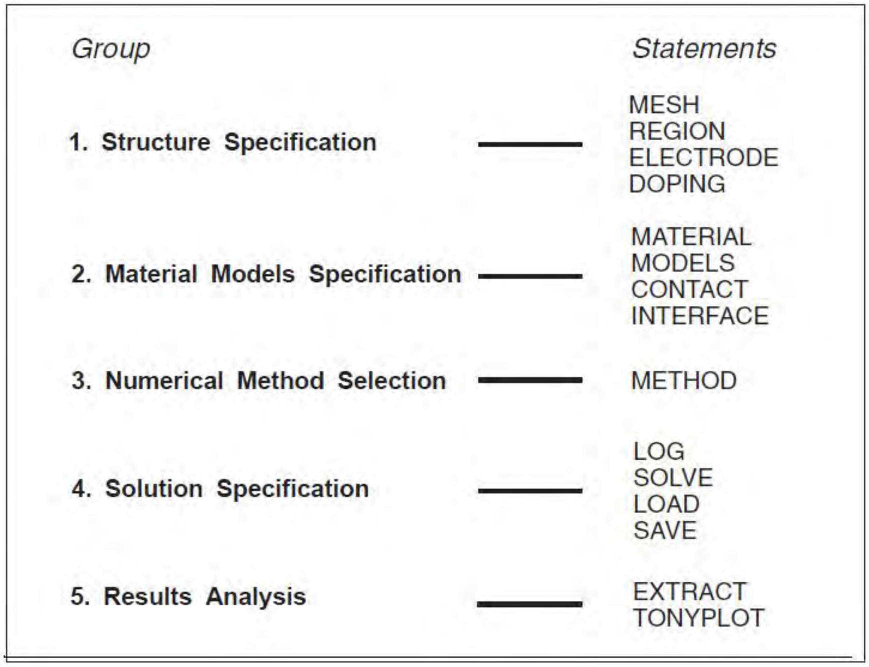
```
go atlas
#
# Solar cell example number 4.
# This example is for a simple Ge
#
# GaInP optical data and layer structure courtesy of IES-UPM, Madrid"
#
# Here we specify a mesh of one square meter ( 100um x 1e10 um).
# Since the structure is essentially one dimensional we ignore
# variations if x and z.  The AUTO keyword indicates that the
# we will not specify mesh lines in the y direction that they
# will be inferred from the REGION statements.
#
mesh auto width=1e10
```

```
EXTRACT> extract name="EQE_ge" max(abs(i."cathode")/elect."source photo current")
initialization has not been performed
EXTRACT> #
EXTRACT> # Plot the results
EXTRACT> #
EXTRACT> quit

ATLAS> tonyplot solarex04.dat
ATLAS> tonyplot solarex04.str -set solarex04.set
ATLAS> quit

ATLAS version 5.16.3.R finished at Wed May 06 16:19:04 2015
```

Figure 20. Screenshot from Deckbuild.



Group	*Statements*
1. Structure Specification	MESH REGION ELECTRODE DOPING
2. Material Models Specification	MATERIAL MODELS CONTACT INTERFACE
3. Numerical Method Selection	METHOD
4. Solution Specification	LOG SOLVE LOAD SAVE
5. Results Analysis	EXTRACT TONYPLOT

Figure 21. Atlas statement hierarchy, from [23, p. 34].

C. BUILDING THE DEVICE

When building the device to be simulated, the first thing to do is to define the mesh that will be the framework for the model. A mesh is a collection of triangles that overlays the device (see Figure 22). Each corner of the triangle is called a node. Nodes are where the equations used to solve for a solution are calculated. When running simulations for a 2D device, there is a limit of 100,000 nodes. A finer mesh results in more triangles, which give more nodes and increases the resolution of the solution to give more accurate results. The downside to this is that the larger number of calculations require an increase in the time it takes to run the simulation. A coarser mesh has the opposite effect. Fewer triangles, less nodes, and less accuracy, but the run time is shorter. The key to setting up a mesh is to recognize where in the device things are happening (such as at a junction). In these areas a finer mesh is desired so that information is not lost. If the triangles are too big, and the characteristics of the device changes too much, then the calculations will not reflect actual conditions. This is akin to the sampling rate for digital circuits. For too slow a sampling rate, the signal being sampled is not accurately captured.

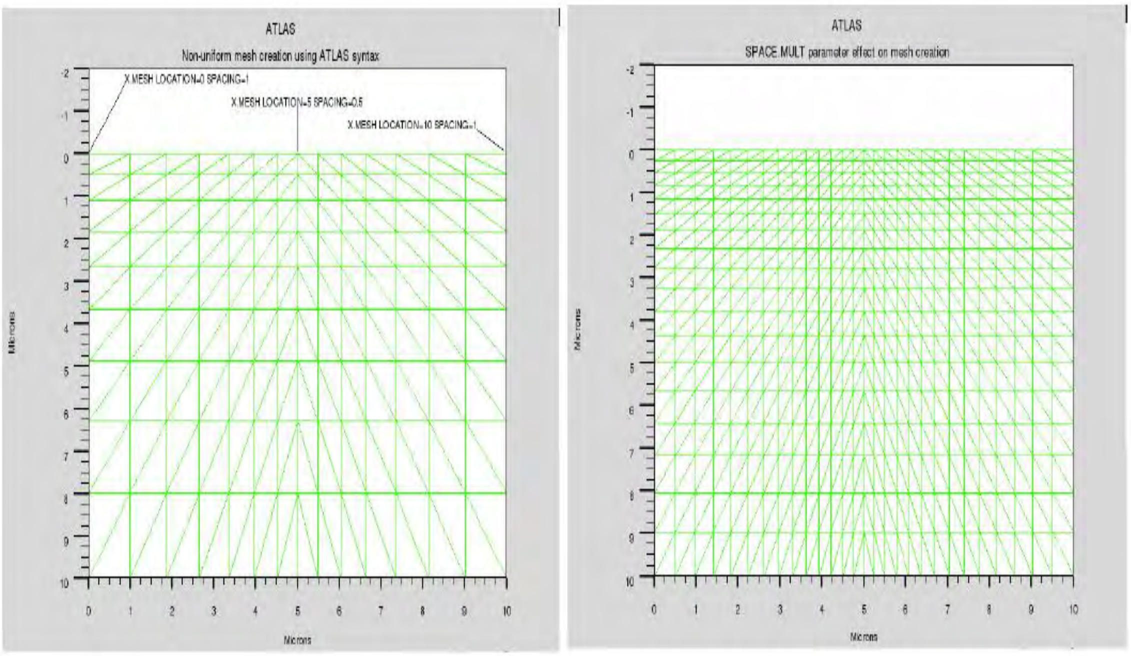

ATLAS
Non-uniform mesh creation using ATLAS syntax
X.MESH LOCATION=0 SPACING=1
X.MESH LOCATION=5 SPACING=0.5
X.MESH LOCATION=10 SPACING=1
Microns
-2
-1
0
1
2
3
4
5
6
7
8
9
10
Microns
ATLAS
SPACE.MULT parameter effect on mesh creation
Microns
Microns

Figure 22. Mesh examples, from [23, p. 38].

1. Defining the Structure

There are three ways a structure can be made. The first, is it can be read in by a previously created structure that has been saved in a separate file. This is done by typing the command line:

```
MESH INFILE=<filename>
```

The second way to define a structure is to use the automatic interface feature of Deckbuild. The automatic interface feature allows a user to run Athena or Devedit from within Deckbuild using the "*Go Athena*" (or "*Go Devedit*") command input. Once the structure is built, run Atlas to run the simulation. This is all done within the same window. The automatic part is due to the handoff of structure information between Athena/Devedit and Atlas. The third way to define a structure is to simply use the command language in Deckbuild to define the structure. The command language for this is:

```
MESH SPACE.MULT=<value>
```

Space.mult is a scaling factor that controls the granularity of the mesh. The default value is one. Any value greater than one gives a coarser mesh, and a value less than one results in a finer mesh. The next statements define where mesh lines in the *x* and *y* direction will be:

```
X.MESH LOCATION=<value> SPACING=<value>

Y.MESH LOCATION=<value> SPACING=<value>
```

These two statements specify the location of the line and the spacing in microns. At least two *x* and two *y* statements are needed to define the mesh. If a different spacing is used, Deckbuild automatically inserts extra lines to allow for a gradual transition. An alternate method to the space.mult command is to use the automatic mesh function:

```
MESH AUTO
```

This statement is followed by *x.mesh* statements. When using the automatic mesh there is no need to use the *y.mesh* command inputs because the *y*-axis mesh is automatically determined based on the region statements.

2. Regions

Once the mesh has been defined, the next step is to define the regions. If a region has been imported from Athena or Devedit, this may not be necessary unless more regions are needed. A typical region statement is:

```
REGION NUMBER=<integer> <material_type> <position_parameters>
```

When using the command line, these regions are numbered, starting with one, up to a maximum of 1000 for a device. Should a user misnumber the regions, Deckbuild automatically numbers them with consecutive numbers. Awkward numbering can be confusing when observing results because the region numbers in the code will not match those in the plots. The material type parameter defines the type of material for the region. The properties for many materials are already stored in the program. They can be changed in a material statement further down in the code once the entire mesh is complete and doping concentrations are defined.

The position parameter defines the location in microns of the region. One way to define the location is with *x.min*, *x.max*, *y.min* and *y.max* statements that look like:

```
X.MIN=<location> X.MAX=<location> Y.MIN=<location> Y.MAX=<location>
```

The *x.min* lists the minimum position in the *x*-direction, and *x.max* lists the maximum position in the *x*-direction. The same holds true for the *y*-axis. It should be noted that the *x*-axis starts at zero on the left and is maximum on the right, just like a standard *x*-*y* Cartesian coordinate system. The *y*-axis, however, is opposite what one might expect. The *y.min* starts at zero on the top, and downward is maximum (see Figure 23). It is important to remember the axis convention since many aspects of the program are position dependent and need to be accurate to ensure proper results. It would not be wise to define a light source underneath a solar cell when trying to simulate its output. Likewise, it would prove confusing for others trying to understand the code and simulation if the standard convention is not followed. It is not always necessary to define both a maximum and minimum for a particular direction. Defining only one suffices, and the program extends the region to the dimensions of the mesh. If *x.min*=10 was defined but no *x.max*, then the region starts at 10.0 microns and extend all the way to the maximum *x*-value defined in the mesh. The same thing occurs for the *y*-direction as long as *y.max* was defined for the mesh, which is not the case when using the auto mesh feature.

Another way to express the location of a region is to define it with respect to other regions in the device such as:

```
REGION NUM=4 BOTTOM THICKNESS=10 MATERIAL=SI DONOR=1e17
REGION NUM=5 TOP THICKNESS=$bulk  MATERIAL=SI
```

Region number four is applied to the bottom of the lowest region with a thickness of 10.0 microns. In this statement the material is defined as silicon and is doped with a donor material with the specified concentration. Region five is similarly defined but would be applied to the top of the upper region. The thickness in this case is defined by a variable. The $ in front of bulk is used to reference the variable bulk, which is defined earlier in the code. Utilizing variables to build regions is useful if the size of regions need to be changed. Variables allow defining regions with respect to the size of previous regions, which are themselves defined with variables. When the time to change a size comes (generally the thickness of a region), it can be changed in the variable, which will carry through to the other variables that rely on it. Use of variables are a quick and easy way to make a change and see the results. The other alternative is to go through the code and redefine *y.min* and *y.max* for the region to be changed and all regions that are below it. When defining regions, it is important to know if the newest region overlaps any others, since then that region’s properties overwrite those in the overlapping area (see Figure 24). Overlap applies to regions brought in from Athena or Devedit. Regions read into Deckbuild can be modified using the *modify* parameter in a region statement. Some common parameters that might be modified are *acceptor, donors, material,* and *name*. There are others that can also be used.

3. Electrodes

Once the regions and materials are specified, the electrodes can be defined. There must be a minimum of one electrode (that touches the semiconductor material) defined for the program to work. Up to 100 electrodes can be specified, and if any of the defined electrodes are given the same name, then the electrodes are considered electrically connected. A typical electrode statement is:

```
ELECTRODE NAME=<electrode name> <postion_parameter>
```

If no *y*-coordinate is given, then the electrode is placed at the top of the device. The *top, bottom*, *right* and *left* parameters can be used to define the location as well.

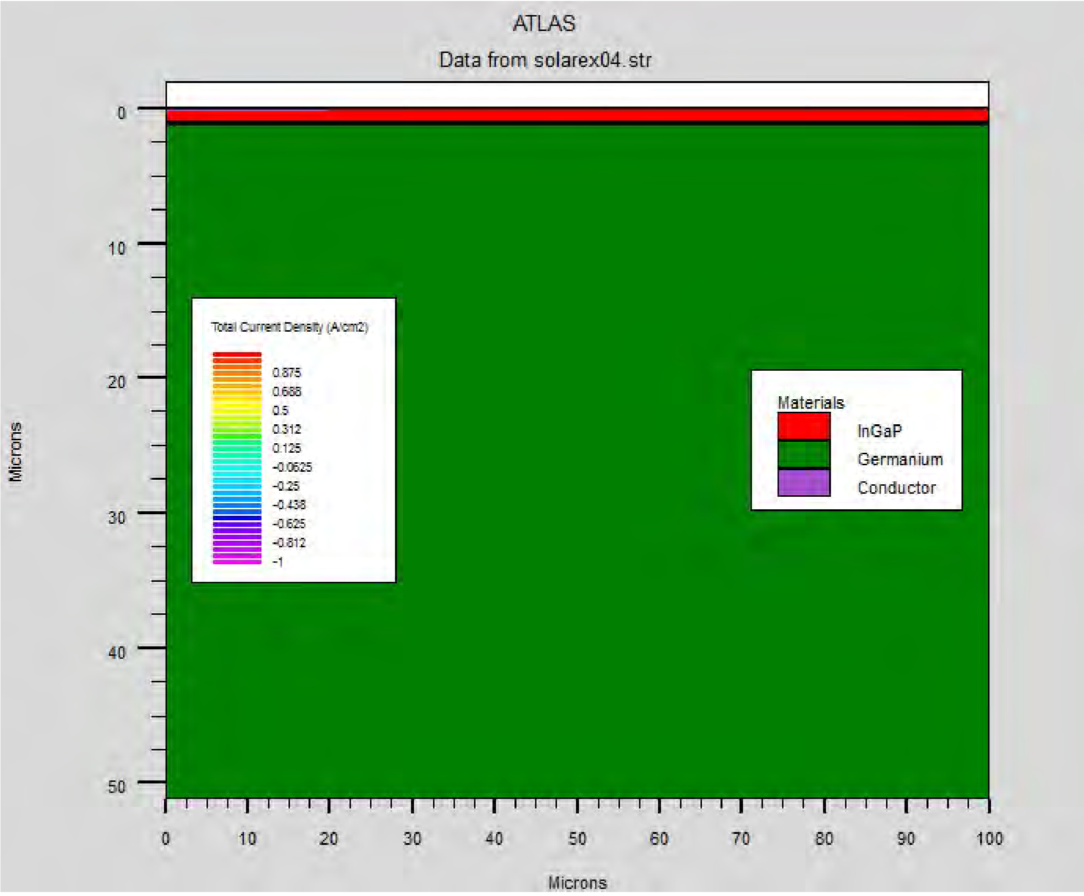


Figure 23. Tonyplot screenshot showing the layout of the *x* and *y*-coordinate system in Atlas.

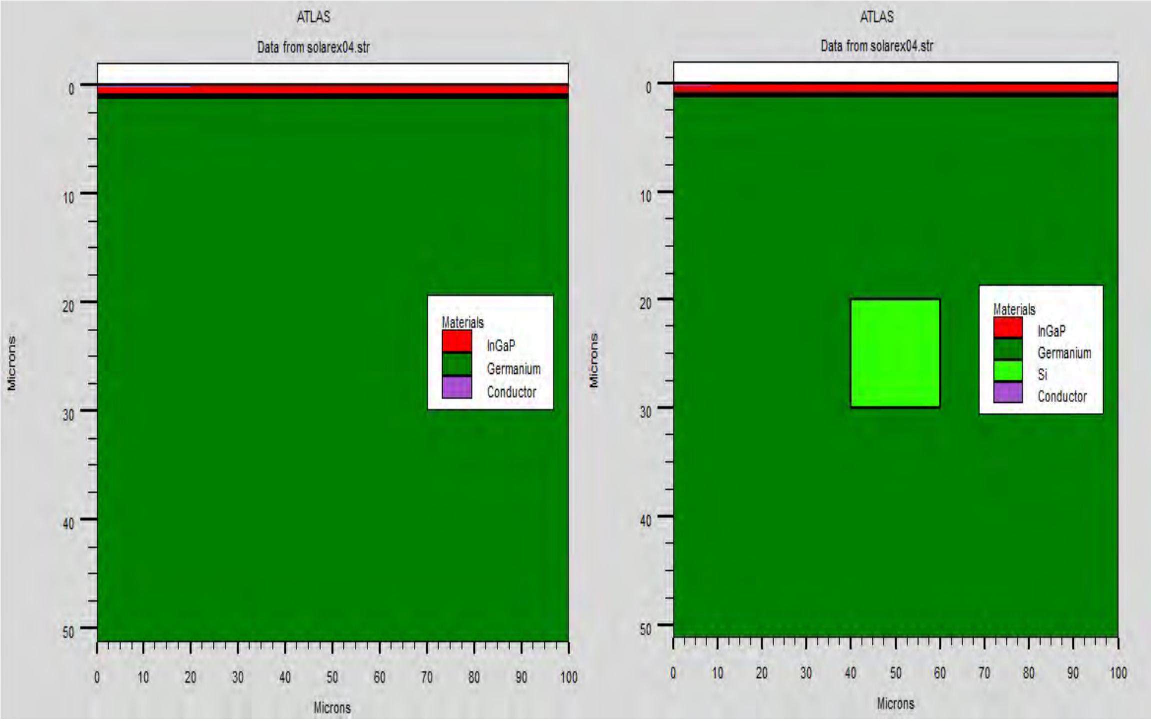


Figure 24. Tonyplot screenshot showing how regions can be overwritten.

4. Doping

Doping parameters are specified next and are generally in the form:

```
DOPING <distribution_type> <dopant_type> <postion_parameter>
```

The distribution type can be uniform, Gaussian, or complementary error function. The dopant type can be specified as *n.type* or *p.type*, and the *concentration* amount can be defined. The position parameter can be specified by region number or the *minimun/maximum* parameters.

5. Modifying Parameters

After the mesh, geometry, and doping parameters have been defined, the characteristics of the electrodes can be changed, the material parameters modified, and the physical models chosen. The electrodes in contact with a semiconductor material are ohmic by default. Defining a work function with the *contact* statement causes it to be treated as a Schottky contact. Using the *materials* statement allows the user to change properties of the specified material such as electron mobility, bandgap, or carrier lifetime.

Specifying *region* after the *material* statement changes the material properties for the material in only the specified region and is useful when multiple regions are made of the same material but not all of them need to be changed. To specify the physical models, use the *models* statement followed by the name of the models. Adding *print* in the *model* statement displays the models and parameters used during the simulation into the runtime output and allows the user to ensure the correct parameters were used.

6. Numerical Solutions

The next thing to define is the numerical solution which tell the program which numerical methods to use when calculating solutions for the device. There are three methods that Deckbuild uses. They are *Gummel*, *Block*, and *Newton* which are specified after the *method* statement. Each numerical method uses a different approach to solving the equations. The *Gummel* method solves for each unknown while keeping the variables constant. It continues doing this until a stable solution is found. The *Newton* method solves the total system of unknowns together, and the *Block* method solves some of the equations the *Gummel* way and the rest the *Newton* way. Generally, the *Newton* method is preferred and is the default if the *method* statement is not used.

7. Obtaining Solutions

To obtain solutions for the created device, the user has the options of using DC, AC, small signal, and transient voltages for the calculations. Once the voltages for the electrodes are defined, Atlas calculates the currents and the internal quantities. To set the voltage of an electrode named anode:

```
SOLVE VANODE=1.0
```

This sets the anode voltage to one volt. Multiple solve statements can be used in succession to ramp up the voltage or a sweep can be used:

```
SOLVE VANODE=0 VSTEP=0.5 VFINAL=5 NAME=anode

SOLVE VSTEP=0.5 VFINAL=10 NAME=anode
```

This sweeps the voltage from zero volts to five volts in one-half volt increments. The second solve statement sweeps from five volts to ten volts in one-half volt increments. Atlas remembers the last voltage of the electrode, five volts after the first solve statement and ten volts after the second solve statement. If an anode voltage was stated in the second solve statement, then the starting voltage for the sweep is instead set to this value. When Atlas attempts to find a solution, the program uses an initial guess for the variables to be evaluated at each bias point. The initial guess usually comes from the two previous solutions, and if none are available convergence problems can arise. Take the following example:

```
SOLVE VGATE=1 VDRAIN=1 VSUBSTRATE=1
```

The voltages of the three electrodes are all set at one volt for the calculations. The program may be able to find a solution, but there is also the possibility that the solutions will not converge. In this case a better approach is:

```
SOLVE VGATE=1

SOLVE VDRAIN=1

SOLVE VSUBSTRATE=1
```

The first solve statement has *vgate* set to one volt, while the other two are at zero volts. The next solve statement sets *vdrain* at one volt with *vgate* remaining at one volt from the previous solve statement and *vsubstrate* remaining at zero volts. The third statement sets the remaining electrode to one volt with the other two remaining at one volt. The third statement solves with all three electrodes at one volt as in the previous example. The difference is that, in this case, Atlas has two previous solutions to use as the initial guess, which reduces the likelihood of convergence problems. When previous solutions are not available, Atlas uses the doping profile to make the initial guess for potential and carrier concentration. To do this the initial solution must be made at zero bias which is done by:

```
SOLVE INIT
```

If this statement is not included, Atlas automatically evaluates an initial solution prior to the first solve statement.

8. Results

When running the simulation it is useful to open a log file to save the terminal characteristics calculated by Atlas. To save the terminal characteristics, open a log file prior to any solve statements:

```
LOG OUTFILE=<filename>
```

This opens a log file and saves it under the chosen filename. All outputs from solve statements are saved in this log file until either another log statement is opened under a different filename or the log is closed by using the log statement with the *off* parameter. Once the simulation is complete, this log file can be used in Tonyplot to view the results. The electrical current values stored in the log file are generally in Amperes per micron, because Atlas is a 2D simulator and sets the *z*-axis to be one micron. If a *width* dimension is specified, the cylindrical coordinate system is used, or when simulating a 3D device, then the units are in Amperes. The log file is also useful for extracting parameters by using the extract command:

```
EXTRACT INIT INF=<filename>
```

This uses the specified file to perform the extraction. If a log file is currently open, then there is no need to specify a file, as the extract statement defaults to the currently open log file. Calculation can be performed on the extracted data and the results saved in a file for later use such as viewing in Tonyplot. The results from the extraction display in the run-time output and are stored in a file called *results.final*. The user can also specify a different file in which to store the results by adding a different file at the end of the extract statement:

```
EXTRACT…….DATAFILE=<filename>
```

Another type of file that provides utility is a solution or structure file, which contains an image of the device at a particular bias point and gives the user the ability to view several parameters of the device. Having the ability to view the device parameters allows a user to see what is going on at that particular point in the simulation. These parameters can greatly assist in troubleshooting the simulation by giving the user an idea of what part of the code to examine for problems.

9. Luminous

Luminous is a general purpose program integrated into Atlas that is used for light propagation and absorption. Luminous calculates the optical intensity within a device and uses this to find the photo-generation rates used in the solution equations. There are four physical models for light propagation. Ray tracing is a general method for resolving non-planar geometries and is generally the preferred starting point but ignores the effects of diffraction and coherence. The transfer matrix method is a 1D method and is recommended for large area devices that includes interference effects. The beam propagation method is a general 2D method that is computationally intensive because it includes diffraction effects. Finally, the finite-difference, time-domain method is a general 2D and 3D method that is the most computationally intensive because it includes both diffraction and coherence effects.

Luminous also has the ability to add complicated AR coatings to a device. The coating is treated as being outside the device and does not interact with the simulation computations. The user can define spherical, ellipsoidal, composite, aspheric, pyramidal, random textured, or user defined surfaces. AR coatings can be a single uniform thickness or consist of multiple layers. To define the optical properties of a coating, use the *interface* statement with the *optical* parameter. An example of an AR coating is:

```
INTERFACE OPTICAL AR.INDEX=2.05 AR.THICK=0.4 REGION=6
```

The *ar.index* parameter defines the refractive index of the layer, while the *ar.thick* sets the thickness to 0.4 microns. Specifying a region indicates the region that the coating is applied to. Alternatively, the *minimum*/*maximum* parameters can also be used.

D. SOLAR CELL SIMULATION

The light propagation and absorption properties of Luminous make it ideal for simulation of solar cells. The user has the ability to define the spectra used in the simulation or to use the built-in spectra for AM0 and AM1.5. To define a beam statement:

```
BEAM NUM=1 AM1.5 WAVEL.START=0.3 WAVEL.END=1.2 \
WAVEL.NUM=50
```

Here the AM1.5 spectrum is specified and is sampled between 0.3 microns and 1.2 microns with 50 samples. The propagation model for the light can also be stated in the beam statement. To get the simulation data, solve statements are used. The first statement is usually:

```
SOLVE B1=1
```

This turns on the light beam. In this example, the value one corresponds to full intensity. Values less than one reduce the intensity and larger values increase the intensity, such as with a solar concentrator. Once the beam is turned on, it is necessary to open a log file to save the current and voltage characteristics and set the voltage of one of the electrodes. Extract statements can also be used to pull pertinent data and solve for common characteristics of solar cells like the fill factor, efficiency, maximum output power, open circuit voltage, and short circuit current, which are shown in the following examples [2]:

```
EXTRACT NAME=“Isc” Y.VAL FROM CURVE(V.”anode,” I.”cathode”) WHERE
X.VAL=0.0

EXTRACT NAME=“Voc” X.VAL FROM CURVE(V.”anode,” I.”cathode”) WHERE
Y.VAL=0.0

EXTRACT NAME=“Pm” MAX(CURVE(V.”anode,” (V.”anode” * I.”cathode”)))

EXTRACT NAME=“Vm” X.VAL FROM CURVE(V.”anode,” (V.”anode”*I.”cathode”) )

WHERE Y.VAL=$”Pm”

EXTRACT NAME=“Im” $”Pm”/$”Vm”

EXTRACT NAME=“FF” $”Pm”/($”Jsc”*$”Voc”)
```

E. DEVEDIT

The Devedit program is a device structure editor. It can be used to create or modify a device that can then be used in Deckbuild. If operating in a Unix environment it runs in a separate Graphical User Interface (GUI) window. To run Devedit simply type *go devedit* in the Deckbuild command window. A useful feature of Devedit is that it gives the user the ability to construct devices with irregular shaped regions. In Deckbuild the regions are more or less limited to rectangular shapes, which limits the utility of the program to make devices. With Devedit this is not an issue as the structure easily transfers to Deckbuild. To make user defined regions, add the *points* parameter in the *region* statement:

```
REGION ID=1 NAME=BULK POINTS="0,0 1,2 4,4 0,4 0,0"
```

which produces the region shown in Figure 25.

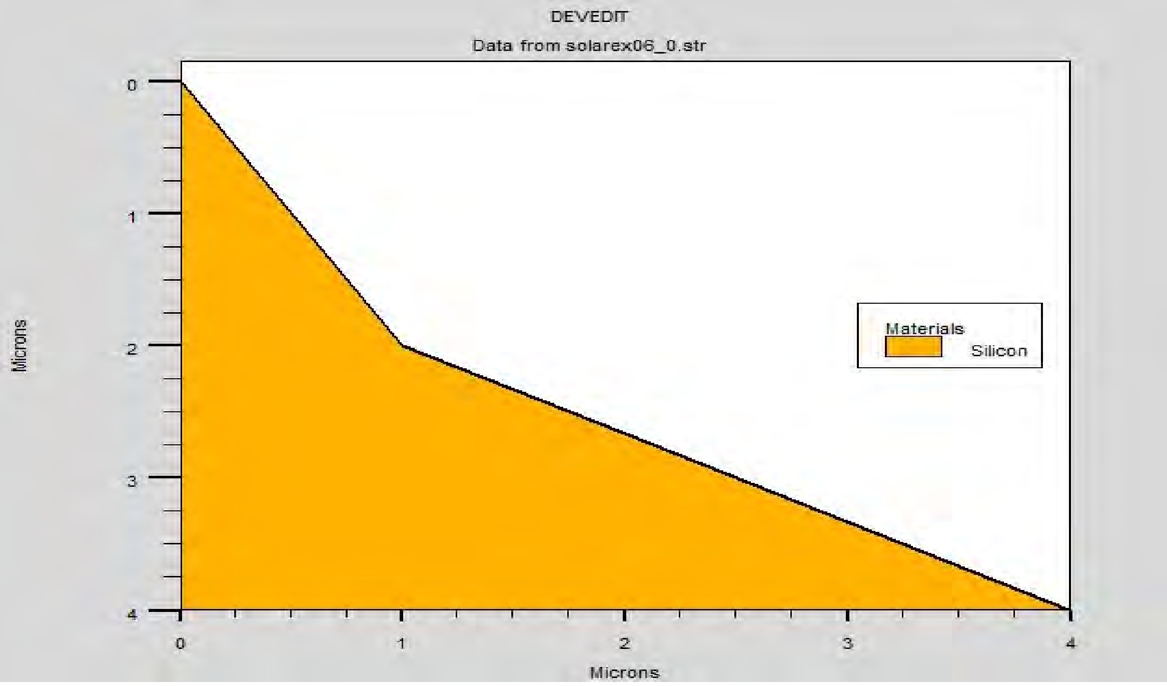


Figure 25. Screenshot showing an irregular structure created in Devedit.

There are other parameters that help with building a device. The *mirror* parameter mirrors the region so that repetitive shapes can be easily reproduced to build the device.

Another parameter is the *stretch* parameter, which allows a region to be made longer, taller, shorter, or narrower. These tools allow very specific devices to be built which can then be simulated in Deckbuild [24].

F. TONYPLOT

Tonyplot is a graphical tool used to plot data obtained from device simulations. It can be called directly in Deckbuild by typing:

```
TONYPLOT <filename> -overlay
```

If the file is a log file, Tonyplot displays the data in an *x*-*y* plot. It the file is a structure file, the information is displayed as a 2D mesh plot.

1. X-Y Plots

When viewing the *x*-*y* plot (see Figure 26), the user has several quantities to choose from to display on the *x*-axis and several to choose from for the *y*-axis. Additionally, the data can be displayed in linear or logarithmic scales. The graph can also be changed to a Cartesian graph, polar plot, or even a Smith chart. With the ruler tool, various aspects of the plot can be measured, such as the slope of a curve, distance between two points, or intercepts. Multiple *x*-*y* plots can be opened in the same window. They can also be overlaid for an easier comparison of data. To do this, select the plots and use the overlay option from the edit menu. Another way is to use the *overlay* option in the command line when calling Tonyplot.

2. 2D Mesh Plot

The 2D mesh plot shows a representation of the device. The user can see the size and shape of different regions, which is useful to make sure the device was constructed as it was intended to be. There are also a variety of options to display information about the characteristics of the device. The mesh plot display in Figure 27 contains buttons that control the various display options. A description of these buttons is shown in Figure 28.

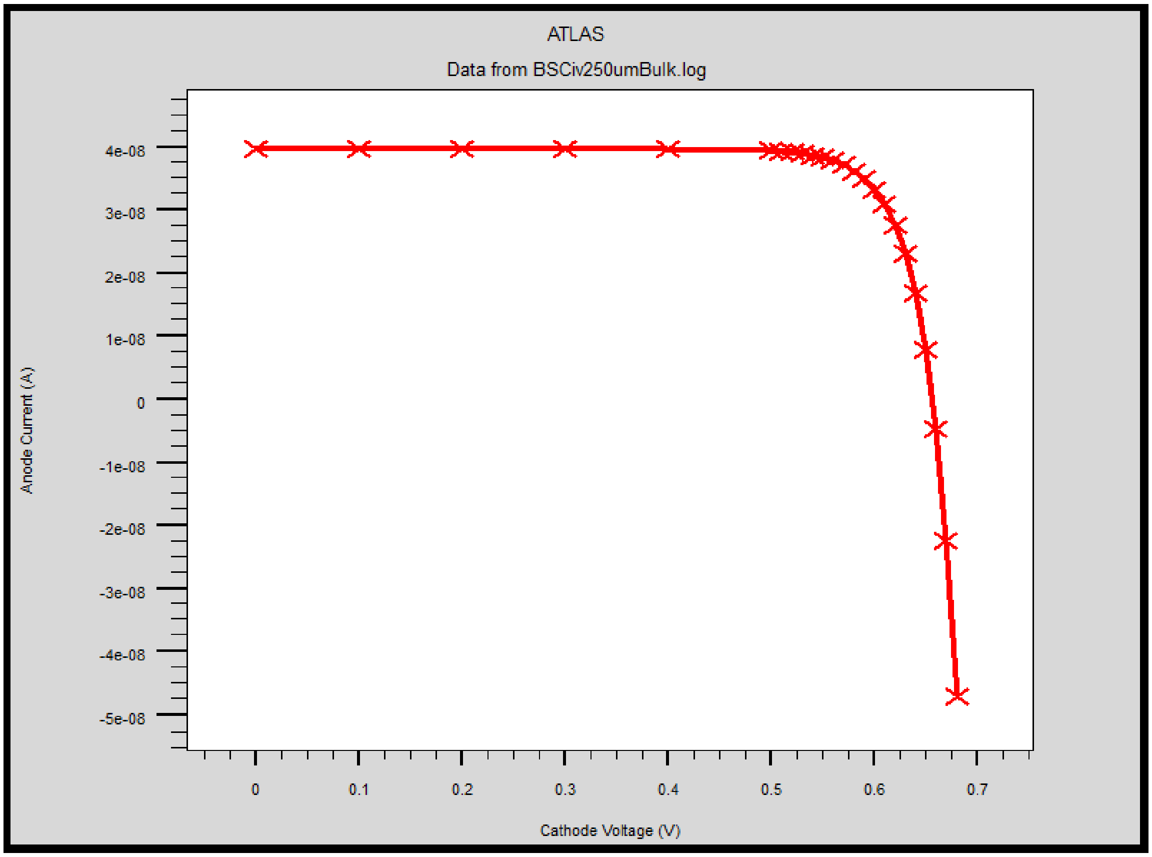


Figure 26. Screenshot of an *x*-*y* plot in Tonyplot.

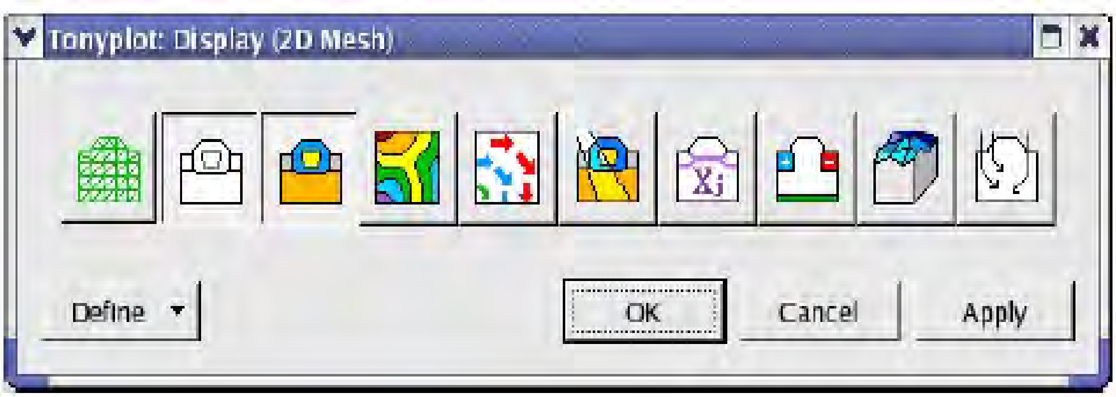
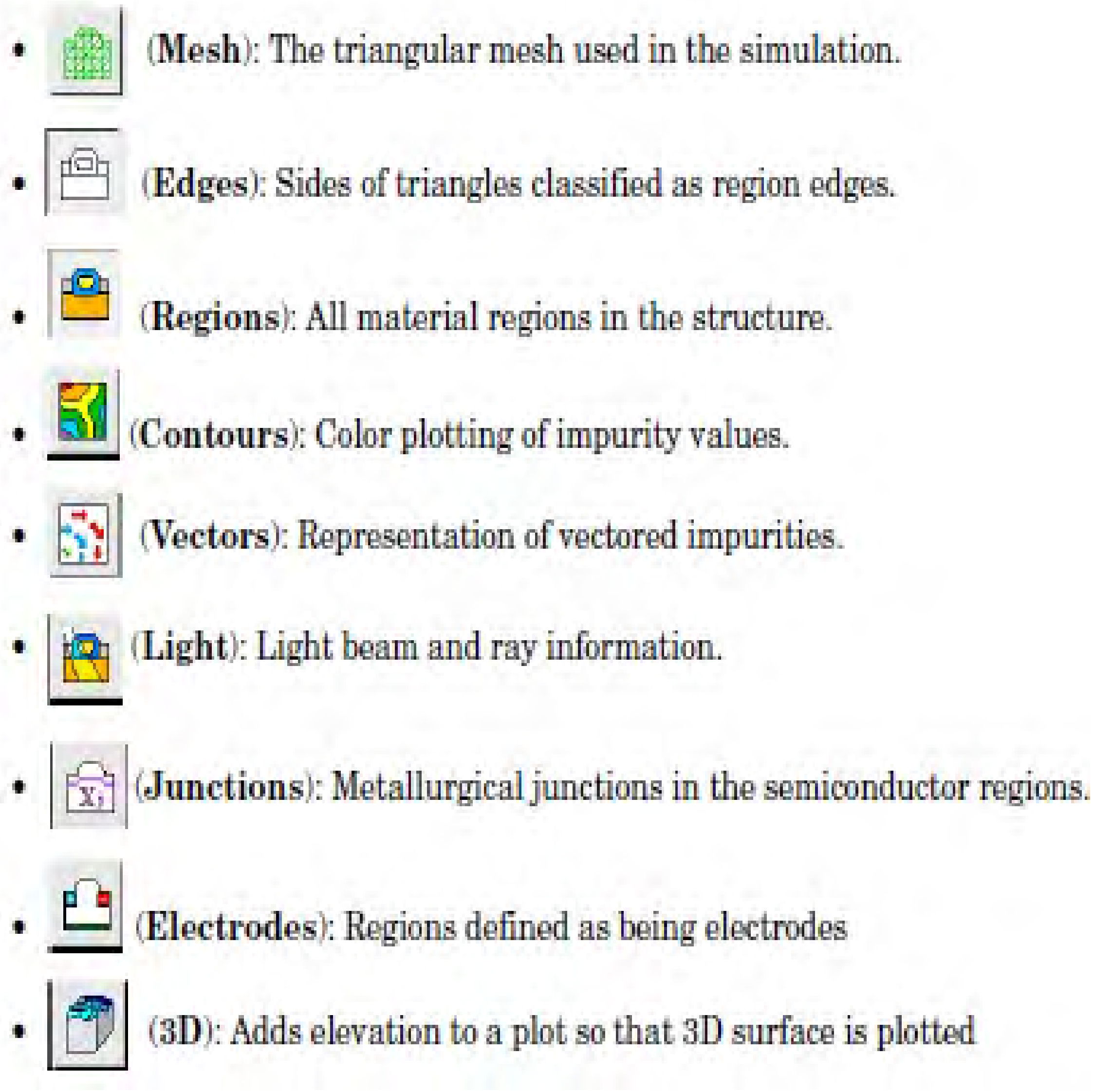


Figure 27. Screenshot showing the mesh plot display buttons for Tonyplot.



- (**Mesh**): The triangular mesh used in the simulation.
- (**Edges**): Sides of triangles classified as region edges.
- (**Regions**): All material regions in the structure.
- (**Contours**): Color plotting of impurity values.
- (**Vectors**): Representation of vectored impurities.
- (**Light**): Light beam and ray information.
- (**Junctions**): Metallurgical junctions in the semiconductor regions.
- (**Electrodes**): Regions defined as being electrodes
- (**3D**): Adds elevation to a plot so that 3D surface is plotted

Figure 28. Description of the display buttons in Tonyplot, after [25, p. 93–94].

The mesh button turns on an overlay of the mesh for the device and shows where all the triangles are located, which gives an indication of areas from which to remove mesh lines, which improves simulation time, or areas to make the mesh finer, which increases accuracy. Another important button is the regions button that turns on the regions in the device, with each region being color coded to the material it is made out of. Placing the mouse cursor over a particular region and pressing the “r” key on the keyboard causes a pop-up window to appear that displays the region name and material.

One of the most useful buttons is the contour button. Turning this on shows a contour plot overlaid onto the device. There are several types of information that the user can choose to display. Some examples are the doping concentration, current density, potential, and optical intensity. Using the cutline tool and cutting a cross-section of the device, we can make a plot of the band diagram. Several 2D mesh files can be opened in the same window. If these plots are all highlighted, there is a movie tool (see Figure 29) that creates a movie consisting of the data from all of the plots. If the plots span a range of voltages, for example, then the movie shows how the contours of the potential change within the device as the voltages change.

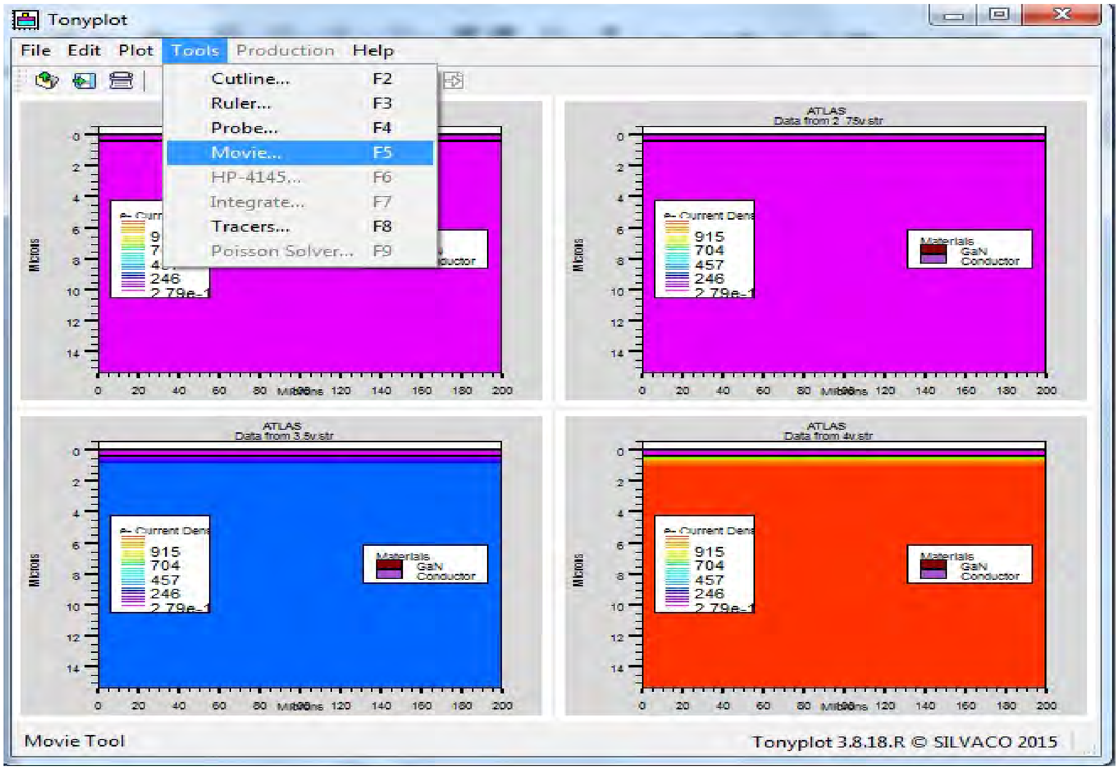


Figure 29. Tonyplot screenshot showing several files opened in the same window with the movie tool selected.

3. Data Management

When viewing the various plots available in Tonyplot, it can become time consuming to adjust the plot settings every time the simulation is run. There is a way to automate this process. Once the plot is set up, there is an option in the file menu to save a set file. The set file is a file that contains the setup parameters for the plot. When running Tonyplot from the command window of Deckbuild, simply add *–set <filename.set>* to cause the plot to open with all the plot settings automatically applied. Sometimes Tonyplot cannot provide the functionality needed with regard to data display. In this case there is a data export function. Simply choose the plot you wish to get the data from and use the export option. A window opens up to allow the user to select the save location, filename, and data type. The data can be saved as a Silvaco format file and can also be saved in a comma separated value format. The latter is a good choice if the data needs to be moved into another data processing program, since this format is widely accepted.

THIS PAGE INTENTIONALLY LEFT BLANK

V. SIMULATION RESULTS

The solar cell model built with Silvaco Atlas software is based on the Sunpower A-300 back-surface-contact cell. The exact makeup of the cell is not available, so the parameters are derived from several different sources. These are all put into the model, which is then simulated to observe the output. The parameters are then varied to alter the output so the cell closely matches the NREL measured values of the A-300.

A. UTILIZING SILVACO

The device structure was initially built using the Devedit component of Atlas. Deckbuild is generally limited to building rectangular shaped regions. Using Devedit allows the construction of non-rectangular textured surfaces. The regions of the device and material type are defined in Devedit. The structure, once built, is then easily loaded into Deckbuild. Other parameters of the model, such as the electron and hole lifetimes, surface recombination velocities, and doping levels are then defined. The top SiO2 passivation layer and AR coating are also added in Deckbuild. This allows the interface parameter to be used, which makes the layers a coating that does not affect the calculations. If the top SiO2 passivation layer is made as a separate region, it greatly increases the number of nodes, which either increases the simulation time or exceeds the maximum node limit, resulting in a program error.

When the device is completely defined, Deckbuild is also used to set the lighting parameters used to illuminate the device and the bias parameters used in the calculations to simulate the device.

B. BUILDING THE CELL

The general structure of the cell matches that of the A-300 shown in Figure 30. The top layer of the cell is coated in a generic AR coating. Below the AR coating is a passivation layer of SiO2. The thickness specification for this layer, as listed in the Cousins patent [18], is one to 15.0 nm. As the thickness of the SiO2 layer is increased, the optical intensity in the cell is reduced. A thickness of 1.0 nm is used in this model to allow more light to enter the cell. The next layer is the FSF, which consists of doped *n*+ material. The thickness of this layer is between 0.01 µm and 5.0 µm [26]. This model uses a thickness of 0.5 µm. The doping concentration of this region is 10^{17} cm^{-3} which is slightly less than the 5.0×10^{18} cm^{-3} recommended by Granek [27]. Increasing the doping value increases V_{oc}, but higher concentrations lower I_{sc}. An FSF value of 10^{17} cm^{-3} gives a good balance between the two and provides a good power output.

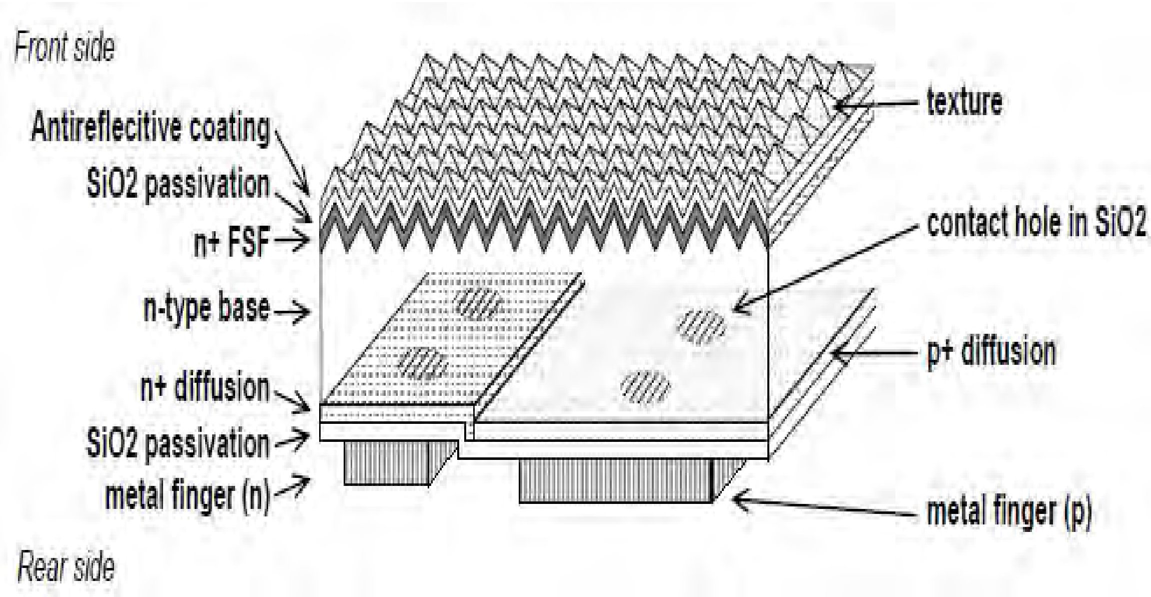


Figure 30. Diagram of the Sunpower A-300 cell, from [1].

The thickest layer in the device is the middle section called the bulk. The thickness was varied between 5.0 and 250.0 µm and simulated at several values to see the effect on the output. The carrier concentration is 10^{15} cm^{-3}, making it lightly doped *n*-type material. The concentration is based on the donor density used in [28]. The surface of this layer is textured with pyramids having a base width of 4.0 µm and an angle normal to the surface of 54.7° [22]. The FSF, SiO2, and AR coatings are all laid on top of the bulk. The bulk is what gives the device its texture to capture more light. The textured portion of the device is seen in Figure 31.

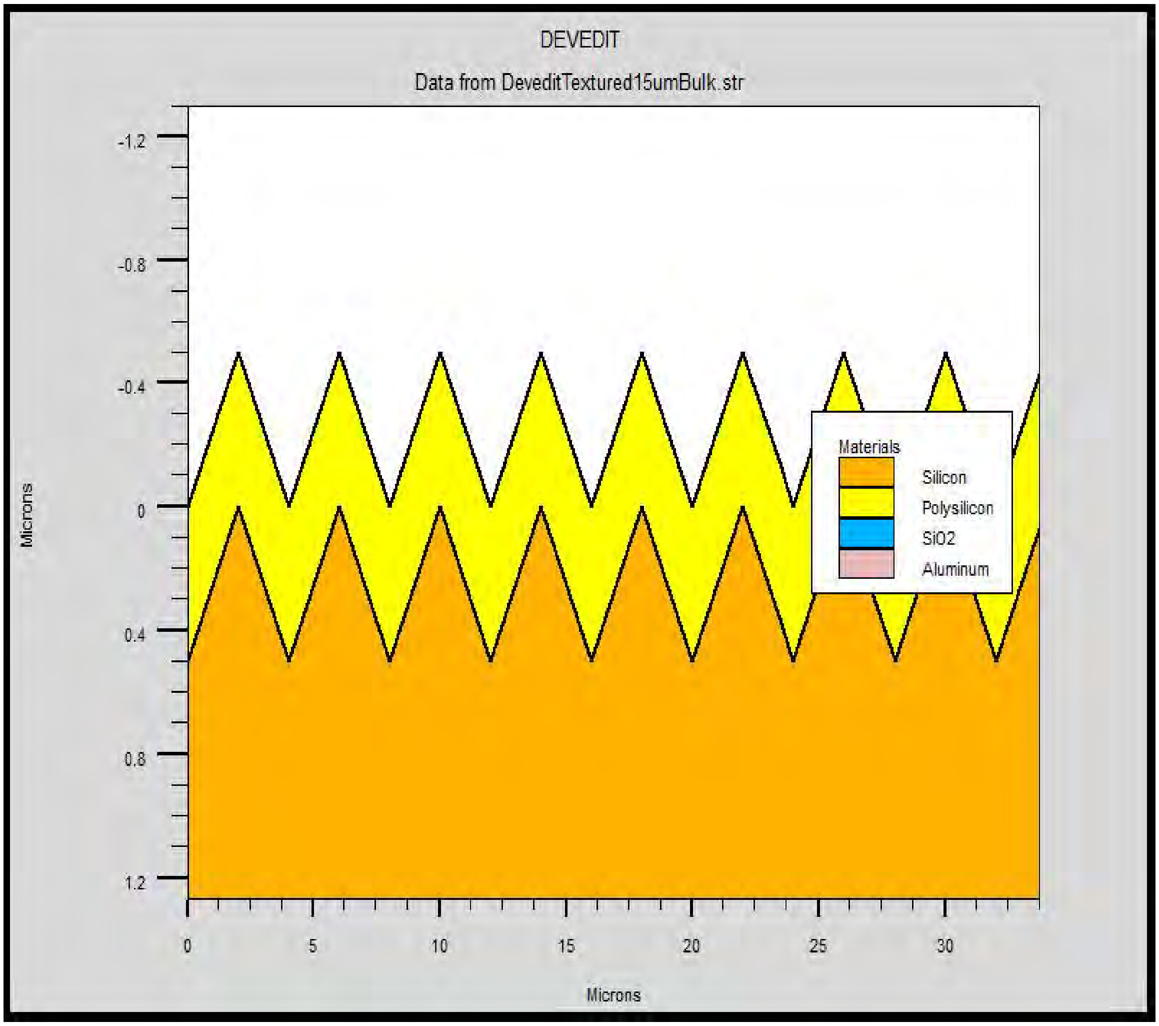


Figure 31. Tonyplot screenshot of the textured surface built in Devedit.

The bottom portion of the cell consists of the *n*+ and *p*+ regions. Cousins' patent [18] lists these regions as having a doping concentration of 10^{20} to 10^{21} . For this model a value in the middle of this range, 5.0×10^{20} was used. The width of the *p*+ region was 60.0 µm and 40.0 µm for the *n*+ region. The *p*+ region is made wider than the *n*+ to reduce the distance the minority carriers have to travel, as discussed in Chapter III.D.4. Below these regions is a layer of SiO2 which is 1.3 µm thick. There are small openings in the SiO2 layer to allow contact between the electrodes and the *p*+/*n*+ regions. These openings are 30.0 µm and 20.0 µm, respectively. The electrodes connected to these two regions are separated by SiO2 regions to prevent shorting between them. A screenshot from Tonyplot shows the complete device in Figure 32.

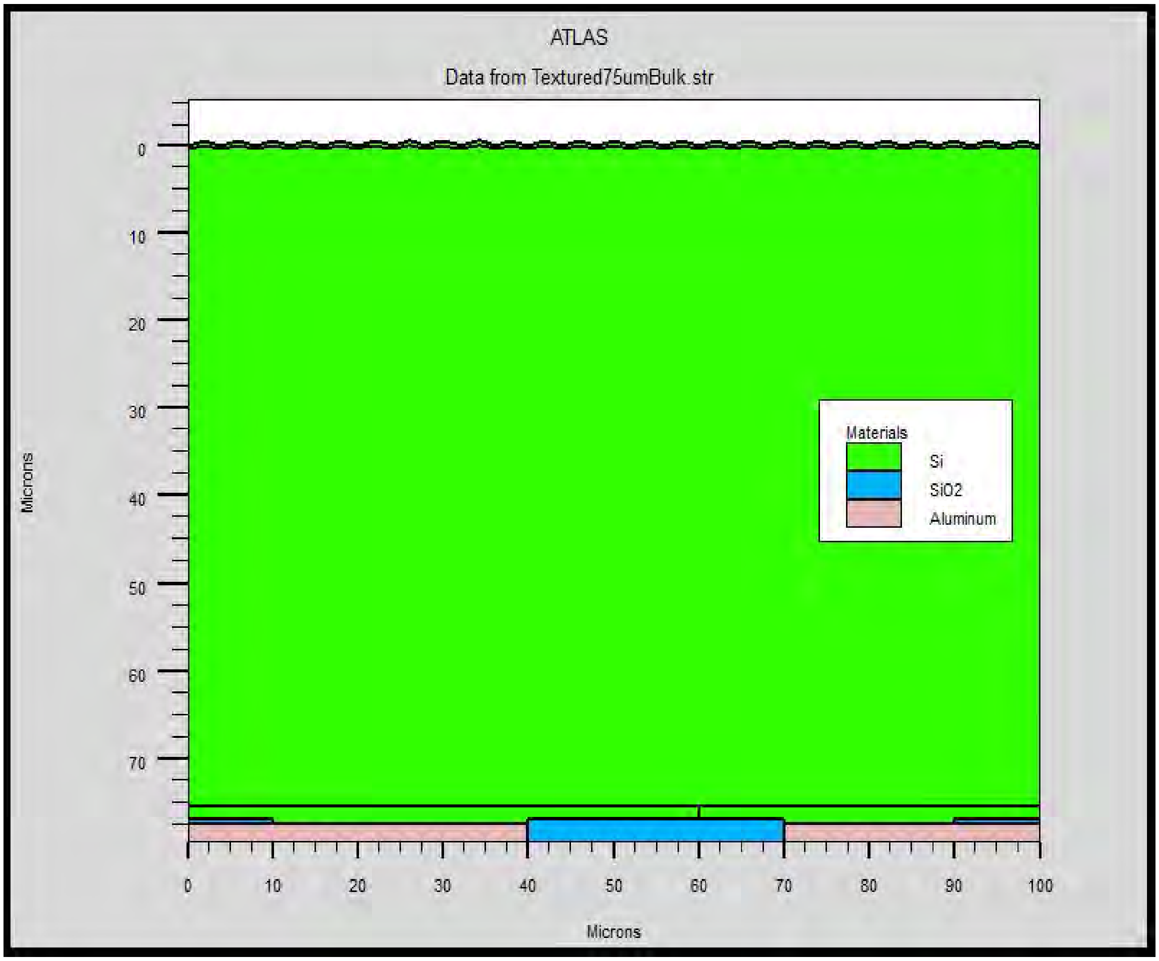


Figure 32. Tonyplot screenshot of the complete device.

The total width of the device is 100.0 µm, which is one pitch of the device. Making it wider to include multiple pitches only increases the number of calculations, which increases the simulation time. One pitch is sufficient to determine the characteristics of the device. The open circuit voltage is the same, and the short circuit current is scaled in proportion to that of a larger device.

C. SIMULATING THE DEVICE

The completed device model underwent several simulation runs, and adjustments to some of the parameters were made in order to improve output power. The IV curve for the final result for a device with a bulk thickness of 100.0 µm is shown in Figure 33.

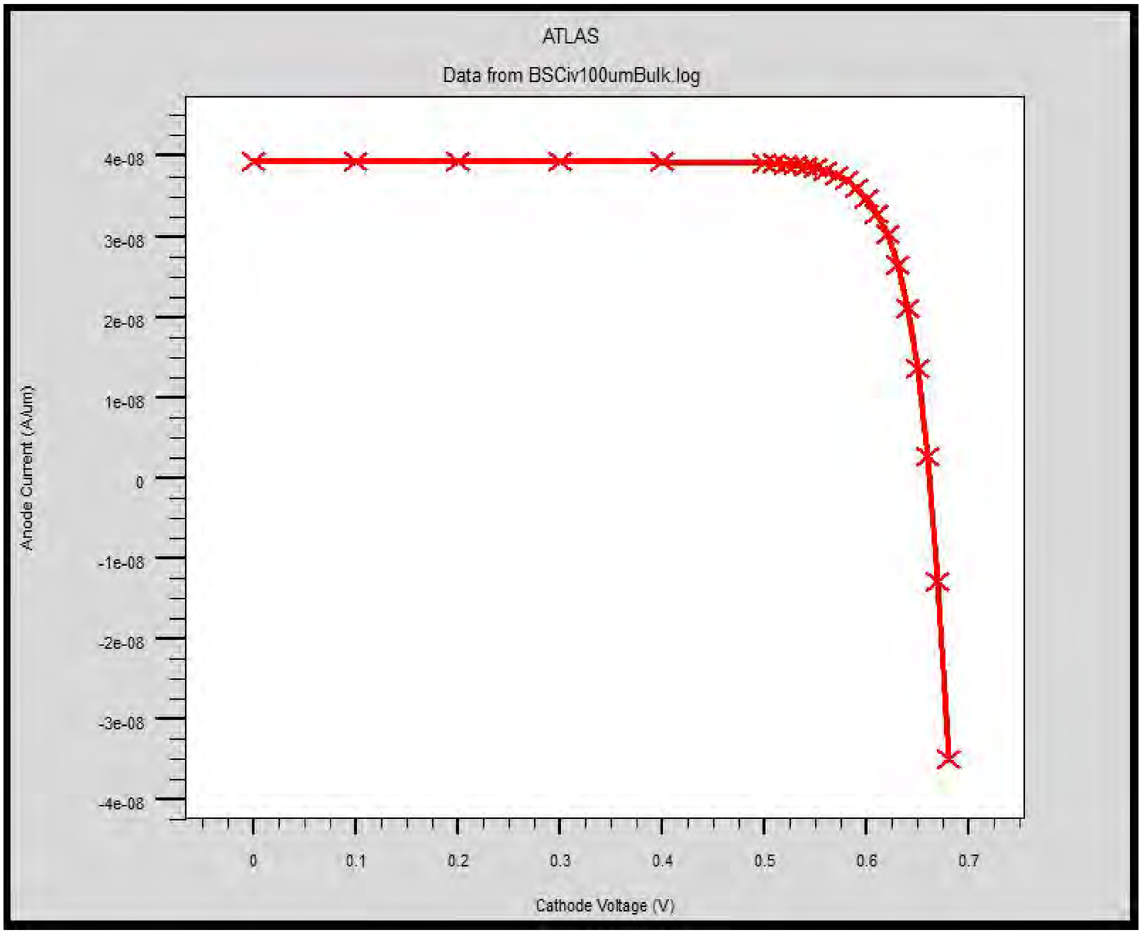


Figure 33. IV Curve for 100µm bulk thickness.

A closer look at the IV curve shows that V_{oc}=653.0 mV, and I_{sc}=3.93×10^{-8} A. To compare this to the NREL measured results, the current must be scaled up based on the size of the device. The A-300 cell is 149.0 cm^2, and the simulated device is 10–^{6} cm^2. To find the scale factor, we divide 149.0 cm^2 by 10–^{6}, giving a factor of 149.0×10^6. Using 149.0×10^6 and multiplying it by 3.97×10^{-8}, we get 5.92 A. The NREL test lists the cell output of 39.5 mA / cm^2 (see Table 1). Multiplying 39.5 mA /cm^2 by the cell area of 149.0 cm^2, we get an output of 5.885 A. The difference between the two values is 35.0 mA. Similarly, the open circuit voltage in the simulation is 660.0 mV, and the NREL measured value is 678.0 mV. The difference between these two values is only 18.0 mV. These results indicate that the model is a fairly accurate representation of the physical device.

Table 1. NREL-reported parameters of low-cost back-contact solar cell at 100 mW/cm^2, AM1.5g, 25° C, from [1].

Area (cm^2)	Silicon	V_{oc} (mV)	J_{sc} (mA/cm^2)	FF	Efficiency (%)
148.9	PV-FZ	678	39.5	0.803	21.5

D. VARYING BULK THICKNESS

To make a cell that is more flexible, it must be thinner. One of the goals of this thesis is to develop a model that can be used to investigate the feasibility of making the cell thinner. A thinner cell has less material for photons to travel through. It is expected the cell will have a drop in power output as the thickness is decreased. If the output stays high enough, this indicates that the cell thickness can be reduced while still providing a high enough output to remain feasible. Multiple simulations of varying bulk thickness show that the cell current output drops as expected (see Figure 34). It does stay fairly consistent until 75.0 µm, meaning the cell can be made thinner and still produce a good output.

When comparing the power output of the different cells, we see that the power does not significantly drop until the bulk decreased to 30.0 µm (see Figure 35). A thickness of 30.0 µm is significantly thinner than the A-300 and would be considerably more flexible. Additionally, looking at the output at 5.0 µm, there is 1.96 W of power output for every cell. Comparing this to SPACES from Chapter II.C, 49 of the 5.0 µm cells at 1.93 W/cell give about 94.5 W of available power. Although the 5.0 µm cell has a significantly degraded output, this is still 32.5 W higher than SPACES. With the higher output, and a thickness that provides a good amount of flexibility, the 5.0 µm cell can be considered as an alternative to SPACES or put to use in other applications requiring thin or flexible solar cells.

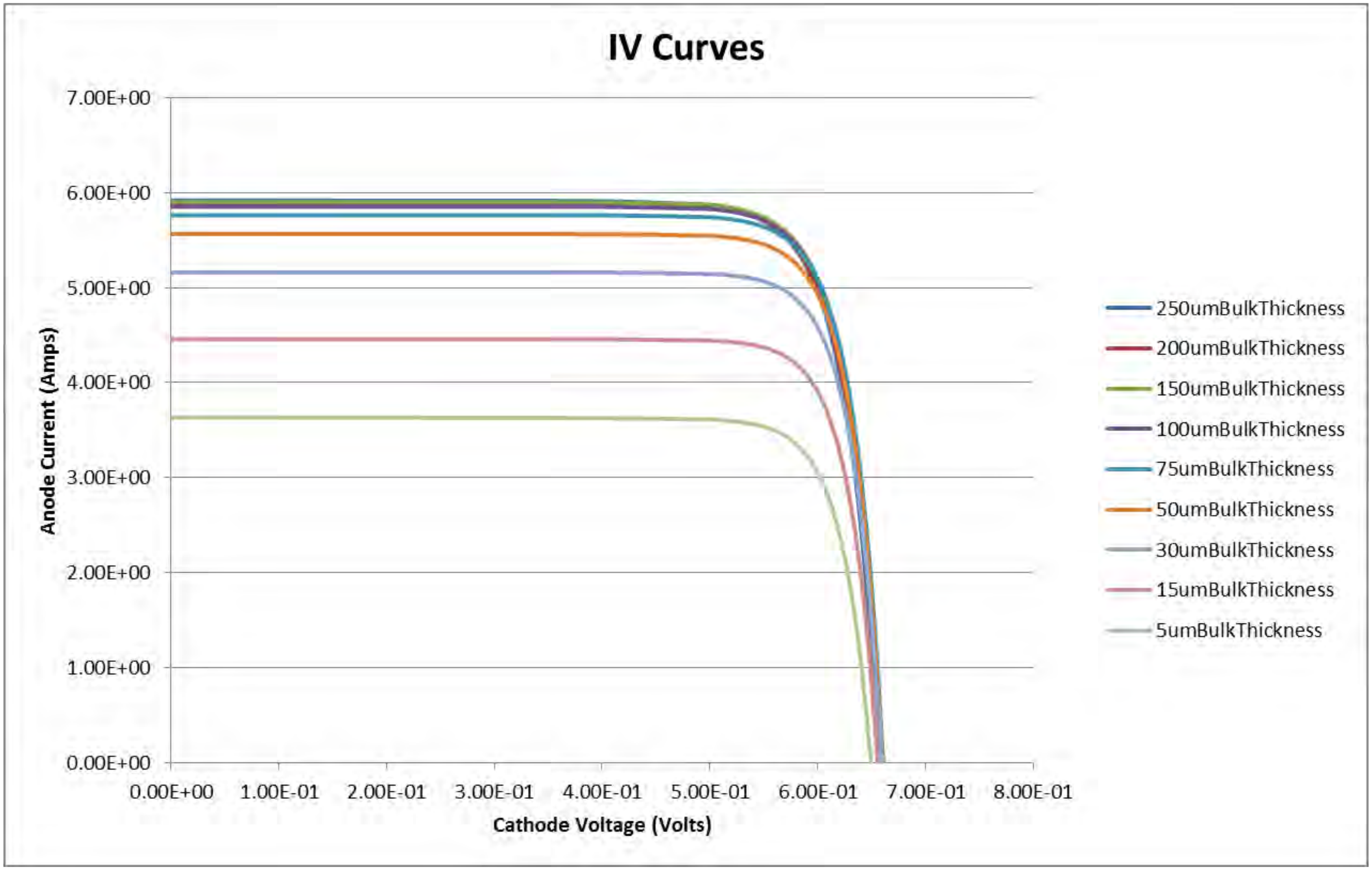


Figure 34. Simulation IV curves for varying bulk thicknesses.

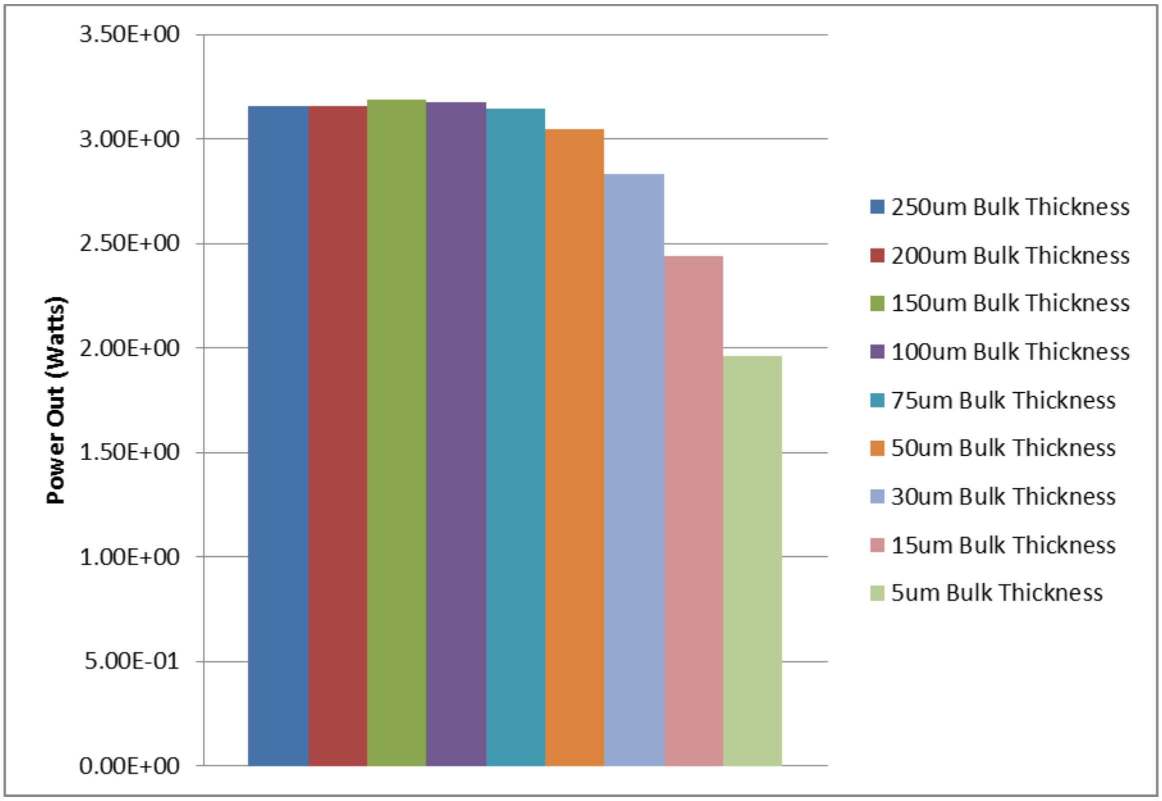


Figure 35. Simulation results of cell power output at maximum power point for varying bulk thicknesses.

VI. CONCLUSIONS AND RECOMMENDATIONS

In this thesis the Sunpower A-300 back-surface-contact solar cell was modeled in order to investigate the feasibility of making the cell thinner for use in military solar blanket applications. The parameters for the cell were taken from a number of different sources and designs. The end result was a working model that approximated the output of the A-300 cell and showed that thinner cells have a high enough power output to be used in military applications.

A. ISSUES

A number of issues had to be solved to achieve a viable simulation. When working with Silvaco, it is best to begin with a simple device and slowly work towards a more complex one. The included examples are always a good start, but if they are used as the basis for the code to be developed, care should be exercised because the examples sometimes contain code that is not needed for a particular simulation. Old code can cause issues to crop up in the development of the new code. Issues occurred when the output current of the Deckbuild structure did not match the current of the Devedit built structure. It took a considerable amount of troubleshooting before it was discovered that the *width* parameter used in the original structure caused Deckbuild to scale the current. This parameter made the output in Amps, while the structure that was built in Devedit had units of Amps per micrometer. When making changes to code, a good rule of thumb is to make small changes so that if a change causes a problem it is easy to identify. If several changes are made at the same time, then it is much more difficult to pinpoint where the problem is, especially if more than one issue arises. While troubleshooting the output current problem just mentioned, it was discovered that the top passivation layer in the Devedit created device was made ten times thicker than it was supposed to be. Not only did this contribute to the output problem, but it complicated the troubleshooting effort. To help make it easier to find problems in the code, it is advisable to save a known working copy as a separate file. Any changes that need to be made can be done in a working file. After several changes are made to the working file, it is different from the original. If a problem crops up with the code or the changes do not have the desired effect, the programmer can easily change the code, or parts of it, back to the original code. The comment character (#) on a line causes Deckbuild to ignore that line, so they can be easily taken in and out of use but left in the code and referred back to if needed. Another option for troubleshooting coding problems is to save and use a structure file that can be viewed in Tonyplot and show many parameters of the device. Early on in the thesis work there were issues with plotting a basic IV curve. Looking at the structure file and viewing the optical intensity revealed that there was an issue with the light source. Focusing on the area in the code where the light source is defined showed that the location for the source had changed from being above the device to inside it. In this way, using the structure file can point to likely areas in the code to look for problems.

B. RECOMMENDATIONS

A back-surface-contact solar cell in Silvaco Atlas software was modeled in this thesis. The design was based on the Sunpower A-300 cell, and the simulation results show that the simulated device was within 18.0 mV and 35.0 mA of the single cell measurements obtained by NREL. The goal was not to precisely duplicate the NREL measured results of the cell but to get a close enough approximation to validate the model. There are two reasons for this. First, there are many parameters that can be manipulated which affect the output, meaning there are millions of different combinations of the device to consider to find the one that matches the A-300. Without having the actual device parameters, it is prohibitively time consuming to find the correct ones. The other reason for not trying to exactly match the A-300 is that this model was made to facilitate future work in this area which will seek to modify the cell.

Comparing the NREL tests to SPACES show that the A-300 is a good candidate for use in a solar blanket type configuration. The A-300 would be better suited though, if it was more flexible, since this would increase durability. Making the cell thinner achieves this. Using the model developed for this thesis, we studied the effects on power output for cell thickness. The preliminary tests show that thinning the cell decreases the output power, but it stays high enough for the cell to remain a viable option. Techniques like genetic algorithms can optimize the performance for a thinner device and hopefully increase the output power from what was observed in this thesis. Another area to study is the use of alternate materials for the cell. Conventional solar cells built with Gallium Arsenide show a better efficiency than silicon based cells. With the advantages that the BSC cell has, utilizing better materials may result in even better performance and also result in a cell thin enough to achieve the desired amount of flexibility.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX. SILVACO ATLAS CODE

A. DEVEDIT CODE

```
go devedit

#This sets up variables used to define the size of the regions.  Using variables makes
#changing the size of regions much simpler.  This is because the bottom regions location
#depend on the position of the regions above.  If each region was defined by coordinates
#then one change would result in changes having to be made to all regions below.  This
#could also lead to error is numbers get mixed up.  The variable allows a user to just
#adjust the one desired parameter and be done.
set FSFlow=.5
set FSFhigh=0
set nbulk=$FSFlow+100
set npPlus=$nbulk+2
set sio2buff=$npPlus-.7
set althick=$npPlus+2

#Sets region one.  The multitude of points is what give the top of the device its texture.
region reg=1 mat=Silicon polygon="0,$FSFlow 2,$FSFhigh 4,$FSFlow \
6,$FSFhigh 8,$FSFlow 10,$FSFhigh 12,$FSFlow 14,$FSFhigh\
16,$FSFlow 18,$FSFhigh 20,$FSFlow 22,$FSFhigh 24,$FSFlow\
26,$FSFhigh 28,$FSFlow 30,$FSFhigh 32,$FSFlow 34,$FSFhigh\
36,$FSFlow 38,$FSFhigh 40,$FSFlow 42,$FSFhigh 44,$FSFlow\
46,$FSFhigh 48,$FSFlow 50,$FSFhigh 50,$nbulk 0,$nbulk"

#This deposits a layer of poly silicon on top of region one.  It has to be a different
#material than what is used in region 1 otherwise the two regions will merge into one.
#The material type is changed in Deckbuild using the modify parameter
deposit start=0 material=poly thickness=.5  region.id=2

#This creates a mirror image of the device.  Region one was defined to 50 microns, using
#mirror it is now the desired 100 microns.  Mirror also simplified the creation of region
#one.  Without it there would have been double the number of points to create the
#texture...ie lots more work and typing
mirror

region reg=3 material=si polygon="0,$nbulk 60,$nbulk 60,$npPlus 0,$npPlus"
region reg=4 material=si polygon="60,$nbulk 100,$nbulk 100,$npPlus 60,$npPlus"
region reg=5 material=sio2 polygon="0,$sio2buff 10,$sio2buff 10,$npPlus 0,$npPlus"
region reg=6 material=sio2 polygon="40,$sio2buff 70,$sio2buff 70,$althick 40,$althick"
region reg=7 material=sio2 polygon="90,$sio2buff 100,$sio2buff\
100,$npPlus 90,$npPlus"
```

```
region reg=8 name=cathode material=aluminum elec.id=1 work.func=0 \
        polygon="0,$npPlus 40,$npPlus 40,$althick 0,$althick"
region reg=9 name=anode material=aluminum elec.id=2 work.func=0 \
        polygon="70,$npPlus 100,$npPlus 100,$althick 70,$althick"

# Set Meshing Parameters
constr.mesh type=Semiconductor default
constr.mesh type=Insulator default
constr.mesh type=Metal default
constr.mesh type=Other default

#This is how you make your mesh more or less dense.  Helpful to cut down sim time or
#increase accuracy
constr.mesh x1=0 y1=.5 x2=100 y2=1.5 min.height=.15 max.height=.25\
min.width=2 max.width=4
constr.mesh x1=0 y1=1.5 x2=100 y2=2 min.height=.25 max.height=.5\
min.width=2 max.width=4
constr.mesh x1=0 y1=3 x2=100 y2=9 min.height=2 max.height=3\
min.width=3 max.width=8
constr.mesh x1=0 y1=10 x2=100 y2=80 min.height=5 max.height=10\
min.width=1 max.width=5
constr.mesh x1=0 y1=100 x2=100 y2=103 min.height=.2 max.height=.3\
min.width=.25 max.width=.5

#Determines the mode that will be used to build the mesh
Mesh Mode=meshbuild

#Saves the structure file so that it can be read into Deckbuild
save type=master file.name=DeveditTextured100umBulk.str

#Plots the structure file.  Was used to ensure the device looked correct prior to bringing it
#into Deckbuild.  It is currently commented out since it is not currently needed
        #tonyplot DeveditTextured100umBulk.str

#Terminates Devedit
quit
```

B. ATLAS CODE

```
go atlas

#Here we bring in the structure file created in Devedit.  The code for Devedit
#could have been included prior to the "go atlas" line and ran in the same window.  It was
#ran separately so that once the structure was built it just had to be read in.  This saved
#sim time from having to build the structure every time the sim was ran
mesh inf=DeveditTextured100umBulk.str

#Here we modify the regions to give them the desired parameters
#This is because in Devedit the deposit command to build the FSF was poly silicon.
#Poly silicon was used to keep this as a separate region.  If Si was used then the two
#regions would have merged into one (because they would be made of the same material)
#and there would be no way to specify the different doping parameters
region num=1 modify material=si donor=1e15 ny=25
region num=2 modify material=si donor=1e17
region num=3  modify material=si acceptor=5e20
region num=4 modify material=si donor=5e20

#The three below commented out lines were used to get a picture of the device before
#calculations This is a good way to ensure the device looks like it should.  They can
#easily be commented in or out as needed The set file is a separate file created with
#Tonyplot.  It simply saves the picture/graph setting so you do not have to set it up every
#time you run the sim

        #save outf=texturecell.str
        #tonyplot texturecell.str -set texturecell.set
        #quit

#Here we specify the material from the Sopra database to capture the index of refraction
material material=si sopra=Sipoly.nk
material material=sio2 sopra=SiO2.nk

#Specify the electron and hole lifetimes for region one.
material region=1 taun0=0.001 taup0=0.001

#Specify the models to be used in the sim
models srh conmob fermi ni.fermi bgn optr auger print.region

#Specify reasonable surface recombination velocities.
#s.s specifies application of interface models at semiconductor-semiconductor interfaces
#s.n electron surf recombination velocity, s.p hole surface recombination velocity. We
#want less than 10 for this model
interface s.s y.min=-.51 y.max=.5 s.n=8  s.p=8
```

```
#The region for the interface optical statement is the region the AR coating is applied to
#The SiO2 is the first layer and the AR coating is the second layer which goes on top of
#the SiO2.  This makes the coatings optical coatings which do not affect the calculations.
#When the SiO2 layer was built in Devedit the number of nodes and triangles exceeded
#the maximum amount allowed in Deckbuild.  With an untextured surface Deckbuild was
#able to use the structure file but the simulation time was much much longer.
interface optical material=sio2 ar.thick=0.001 region=2
interface optical ar.index=2.05 ar.thick=0.4 coating=1 layer=2

#The beam is placed above the device with an AM1.5 intensity.  The spectrum is limited
#to that which the device will respond to, that is 0.3-1.2 microns with 50 sample points in
#between. Since no method is stated the default is Ray Tracing
beam num=1 x.origin=50 y.origin=-2 AM1.5 wavel.start=0.3 wavel.end=1.2\
wavel.num=50

#This solves the zero state solution.  Use to give a good initial guess
solve init

#Solve the beam statement.  The number one sets the beam power to 100%.  Greater than
#one would give more, like when using a solar concentrator, and less than one would
#give less.
solve b1=1

#Save a structure file including optical intensity
output opt.intens
save outf=Textured100umBulk.str

#Plot the structure file just saved
tonyplot Textured100umBulk.str -set Textured.set

#Opens a log file to record results from voltage sweep
log outfile=BSCiv100umBulk.log

#Runs a voltage sweep to step the voltage from 0V to 0.68V.  VCathode only needs to be
#defined in the first solve statement.  In the subsequent solve statements the voltage picks
#with its last value.  So after the first statement the voltage was 0.5 volts, in the second
#statement it starts out at 0.5V
solve vcathode=0.0  vstep=0.1 vfinal=0.5 name=cathode b1=1
solve vstep=0.01 vfinal=0.65 name=cathode b1=1
solve vstep=0.01 vfinal=0.68 name=cathode b1=1
#Turns off the log file
log off
#Plots the results from the log file and terminates the simulation
tonyplot BSCiv100umBulk.log -set iv.set
quit
```

LIST OF REFERENCES

[1] W.P. Mulligan, D.H. Rose, M.J. Cudzinovic, D.M. De Ceuster, K.R. McIntosh, D.D. Smith, and R.M. Swanson, "Manufacture of solar cells with 21% efficiency," Sunpower Corp., Sunnyvale, CA, n.d.

[2] D. McCloy, "High efficiency solar cells: a model in Silvaco," M.S. thesis, Dept. of Electrical and Computer Eng., Naval Postgraduate School, Monterey, CA, 1999.

[3] P. Michalopoulos, "A novel approach for the development and optimization of state-of-the-art photovoltaic devices using Silvaco," M.S. thesis, Dept. of Electrical and Computer Eng., Naval Postgraduate School, Monterey, CA, 2002.

[4] K. Fotis, "Modeling and simulation of a dual-junction CIGS solar cell using Silvaco Atlas," M.S. thesis, Dept. of Electrical and Computer Eng., Naval Postgraduate School, Monterey, CA, 2012.

[5] Iris Technologies, "SPACES generation II (SPA II) solar portable alternative communications energy system," Iris Tech., Irvine, CA, Nov. 2013.

[6] Sunpower Corporation, "A-300 solar cell specification sheet," Sunpower Corp., Sunnyvale, CA, Mar. 2005.

[7] A.R. Harvey, "The modification of homer software application to provide the United States Marine Corps with an energy planning tool," M.S. thesis, Dept. of Electrical and Computer Eng., Naval Postgraduate School, Monterey, CA, 2012.

[8] R.A. Serway, and J.W. Jewett Jr., *Physics for Scientists and Engineers,* 6th ed. Belmont, CA: Brooks/Cole-Thompson Learning, 2004, pp. 709, 1193.

[9] B. Streetman, *Solid State Electronic Devices,* 4th ed. Englewood Cliffs, NJ: Prentice Hall, 1995, pp. 56, 439.

[10] "Solar Energy," class notes for EC3240: Renewable Energy at Military bases and for the Warfighter, Dept. of Electrical and Computer Engineering, Naval Postgraduate School, Monterey, CA, summer 2014.

[11] B. Garcia Jr., "Indium Gallium Nitride multijunction solar cell simulation using Silvaco atlas," M.S. thesis, Dept. of Electrical and Computer Eng., Naval Postgraduate School, Monterey, CA, 2007.

[12] Cool Roofing Materials Database. Object name SPECTRUM OF SOLAR RADIANCE. [Online]. Available: http://energy.lbl.gov/coolroof/intro.htm. Accessed May 09, 2015.

[13] X.F. He. (2010, Feb. 01). Uniquely challenging. [Online]. Available: http://www.vision-systems.com/articles/2010/02/uniquely-challenging.html

[14] Maximum power point tracking. (n.d.). Peacock Media Group. [Online]. Available: https://www.solarquotes.com.au/inverters/mppt/. Accessed May 09, 2015.

[15] Photovoltaics. (n.d.). Engineering.com. [Online]. Available: http://www.engineering.com/SustainableEngineering/RenewableEnergyEngineering/SolarEnergyEngineering/Photovoltaics/tabid/3890/Default.aspx. Accessed May 09, 2015.

[16] Maximum power point (MPP). (n.d.). The worlds of David Darling. [Online]. Available: http://www.daviddarling.info/encyclopedia/M/AE_maximum_power_point.html. Accessed May 09, 2015.

[17] Sunpower Corporation, "More power. Better investment," Sunpower Corp., San Jose, CA, 2008.

[18] P.J. Cousins, "Solar cell having doped semiconductor heterojunction contacts," U.S. Patent 8,815, 631 B2, Aug. 26, 2014.

[19] F.F. Masouleh, and N. Das, "Application of metal-semiconductor-metal photodetector in high-speed optical communication systems," in *Advances in Optical Communications*, N. Das, Ed., InTech, 2014, p. 94.

[20] C. Honsberg, and S. Bowden. (n.d.). Light trapping. [Online]. Available: http://www.pveducation.org/pvcdrom/design/light-trapping. Accessed May 09, 2015.

[21] M. Aurientis, "Interdigitated back contact n-type solar cell with black silicon anti-reflecting layer: simulations and experiments," M.S. thesis, Dept. Micro-and Nanosciences, Aalto Univ., Espoo, Finland, 2014.

[22] K. Mangersnes, "Back-contact back-junction silicon solar cells," Ph.D. dissertation, Dept. of Physics, Univ. of Oslo, Oslo, Norway, 2010.

[23] *Atlas User's Manual*, Silvaco International, Santa Clara, CA, 2005.

[24] *Devedit User's Manual*, Silvaco International, Santa Clara, CA, 2014.

[25] *Tonyplot User's Manual*, Silvaco International, Santa Clara, CA, 2014.

[26] S.B. Rim, M. Morse, T. Kim, and M.J. Cudzinovic, "Thin silicon solar cell and method of manufacture," U.S. Patent 8,822,257 B2, Sep. 2, 2011.

[27] F. Granek, “High-efficiency back-contact back-junction silicon solar cells,” Ph.D. dissertation, Fraunhofer Institute for Solar Energy Systems, Freiburg im Breisgau, Germany, 2009.

[28] M. Filipic, F. Smole, and M. Topic, “Optimization of interdigitated back contact geometry in silicon heterojunction solar cell,” in *Intl. Conf. on Numerical Simulation of Optoelectronic Devices*, Palma de Mallorca, Spain, 2014, pp. 161–162.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California